Package ‘ridigbio’

October 1, 2024

Title Interface to the iDigBio Data API

Version 0.4.1

Date 2024-9-27

Encoding UTF-8

Description An interface to iDigBio's search API that allows downloading specimen records. Searches are returned as a data.frame. Other functions such as the metadata end points return lists of information. iDigBio is a US project focused on digitizing and serving museum specimen collections on the web. See <`https://www.idigbio.org`> for information on iDigBio.

URL `https://github.com/iDigBio/ridigbio`, `https://idigbio.github.io/ridigbio/`

BugReports `https://github.com/iDigBio/ridigbio/issues`

Depends R (>= 3.0.1)

Imports stats, plyr, httr, jsonlite, leaflet, kableExtra, tidyverse, cowplot

Suggests testthat, markdown, knitr, utils, rmarkdown

VignetteBuilder knitr

Repository CRAN

RoxygenNote 7.3.1

NeedsCompilation no

Author Francois Michonneau [aut, cph] (Original Author),
Matthew Collins [aut] (Original Author),
Scott Chamberlain [ctb],
Kevin Love [ctb],
Hem Nalini Morzaria-Luna [ctb],
Michelle L. Gaynor [ctb, aut],
Jesse Bennett [cre] (Maintainer)

Maintainer Jesse Bennett <`j3553@ufl.edu`>

Date/Publication 2024-10-01 02:10:03 UTC

Contents

`build_field_lists` *Build fields and fields_exclude for queries.*

Description

Given the desired fields to be returned, intelligently add an exclusion for the data array if warranted and handle the "all" keyword. And do so without setting both fields and fields_exclude due to fact that the API will return wrong results if are passed. This is still posssible if the user deliberately sets both. Not exported.

Usage

```
build_field_lists(fields, type)
```

Arguments

`fields`	character vector of fields user wants returned
`type`	type of records to get fields for

Value

list list with fields key for df fields and query key for parameters to be merged with the query sent

---

`idig_build_attrib` *Attribution dataframe of iDigBio records query*

Description

Function to build attribution dataframe from a query to the iDigBio API

Usage

```
idig_build_attrib(dat)
```

Arguments

`dat`	dataframe generated by idig_search method

Details

This function differs from the attribution metadata that is attached to the dataframe returned by the idig_search_* methods. It summarizes the record sets used by records in the dataframe, not the record sets that have records that match the query sent to iDigBio. This is useful if only part of the records for a query are downloaded, for example with the limit and offset parameters.

Exported.

Value

a data frame

Author(s)

Kevin Love

`idig_check` *check HTTP code*

Description

Checks for HTTP error codes and JSON errors.

Usage

```
idig_check(req)
```

Arguments

`req`	the returned request

Details

Part 1 of the error checking process. This part handles HTTP error codes and then calls part 2 which handles JSON errors in the responses. Not exported.

Value

nothing. Stops if HTTP code is >= 400

Author(s)

Francois Michonneau

`idig_check_error` *Check is the request returned an error.*

Description

Checks for error messages that can be returned by the API in JSON.

Usage

```
idig_check_error(req)
```

Arguments

`req`	the returned request

Details

Part 2 of the error checking process. Checks the JSON response for error messages and stops if any are found. Not exported.

Value

nothing. Stops if request contains an error.

Author(s)

Francois Michonneau

`idig_count_media` *Count media endpoint*

Description

Count media records matching a query.

Usage

```
idig_count_media(rq = FALSE, mq = FALSE, ...)
```

Arguments

`rq`	iDigBio record query in nested list format
`mq`	iDigBio media query in nested list format
`...`	additional parameters

Details

Quickly return a count of the media records matching the query(s) provided.

Value

count of media records matching the query(s)

Author(s)

Matthew Collins

`idig_count_records` *Count record endpoint*

Description

Count specimen records matching a query.

Usage

```
idig_count_records(rq = FALSE, ...)
```

Arguments

`rq`	iDigBio record query in nested list format
`...`	additional parameters

Details

Quickly return a count of the specimen records matching the query(s) provided.

Value

count of specimen records matching the query(s)

Author(s)

Matthew Collins

`idig_GET` *internal GET request*

Description

Internal function for GET requests.

Usage

```
idig_GET(path, ...)
```

Arguments

`path`	endpoint
`...`	additional arguments to be passed to httr::GET

Details

Generates a GET request and performs the checks on what is returned. Not exported.

Value

the request (as a list)

Author(s)

Francois Michonneau

`idig_meta_fields` *meta fields endpoint*

Description

List of fields in iDigBio.

Usage

```
idig_meta_fields(type = "records", subset = FALSE, ...)
```

Arguments

`type`	string type of fields to return, defaults to "records"
`subset`	set of fields to return, "indexed", "raw", or unset for all
`...`	additional parameters

Details

Return a list of media or specimen fields that are contained in iDigBio.

Value

list of fields of the requested type

Author(s)

Matthew Collins

`idig_parse` *parse successfully returned request*

Description

Parses output of successful query to return a list.

Usage

```
idig_parse(req)
```

Arguments

`req`	the returned request

Details

Not exported.

Value

a list

Author(s)

Francois Michonneau

`idig_POST` *internal POST request*

Description

Internal function for POST requests.

Usage

```
idig_POST(path, body, ...)
```

Arguments

`path`	endpoint
`body`	a list of parameters for the endpoint
`...`	additional arguments to be passed to httr::POST

Details

Generates a POST request and performs the checks on what is returned. Not exported.

Value

the request (as a list)

Author(s)

Francois Michonneau

`idig_search` *Basic searching of iDigBio records*

Description

Base function to query the iDigBio API

Usage

```
idig_search(
  type = "records",
  mq = FALSE,
  rq = FALSE,
  fields = FALSE,
  max_items = 1e+05,
  limit = 0,
  offset = 0,
  sort = FALSE,
  ...
)
```

Arguments

`type`	string type of records to query, defaults to "records"
`mq`	iDigBio media query in nested list format
`rq`	iDigBio record query in nested list format
`fields`	vector of fields that will be contained in the data.frame
`max_items`	CURRENTLY IGNORED, SEE ISSUE #33 maximum number of results allowed to be retrieved (fail-safe)
`limit`	maximum number of results returned
`offset`	number of results to skip before returning results
`sort`	vector of fields to use for sorting, UUID is always appended to make paging safe
`...`	additional parameters

Details

This function is wrapped for media and specimen record searches. Please consider using `idig_search_media` or `idig_search_records` instead as they supply nice defaults to this function depending on the type of records desired.

Fuller documentation of parameters is in the `idig_search_records` function's help.

Exported to facilitate wrapping this package in other packages.

Value

a data frame

Author(s)

Francois Michonneau

Examples

```
## Not run:
# Ten media records related to genus Acer specimens
idig_search(type="media", rq=list(genus="acer"), limit=10)

## End(Not run)
```

`idig_search_media` *Searching of iDigBio media records*

Description

Function to query the iDigBio API for media records

Usage

```
idig_search_media(
  mq = FALSE,
  rq = FALSE,
  fields = FALSE,
  max_items = 1e+05,
  limit = 0,
  offset = 0,
  sort = FALSE,
  ...
)
```

Arguments

`mq`	iDigBio media query in nested list format
`rq`	iDigBio record query in nested list format
`fields`	vector of fields that will be contained in the data.frame, defaults to "all" which is all indexed fields
`max_items`	maximum number of results allowed to be retrieved (fail -safe)
`limit`	maximum number of results returned
`offset`	number of results to skip before returning results
`sort`	vector of fields to use for sorting, UUID is always appended to make paging safe
`...`	additional parameters

Details

Also see `idig_search_records` for the full examples of all the parameters related to searching iDigBio.

Wraps `idig_search` to provide defaults specific to searching media records. Using this function instead of `idig_search` directly is recommened. Record queries and media queries objects are allowed (rq and mq parameters) and media records returned will match the requirements of both.

This function defaults to returning all indexed media record fields.

Value

A data frame with fields requested or the following default fields:

- accessuri
- datemodified: Date last modified, which is assigned by iDigBio.
- dqs: Data quality score assigned by iDigBio.
- etag: Tag assigned by iDigBio.
- flags: Data quality flag assigned by iDigBio.
- format
- hasSpecimen: TRUE or FALSE, indicates if there is an associated record for this media.
- licenselogourl
- mediatype: Media object type.
- modified
- recordids: List of UUID for associated records.
- records: UUID for the associated record.
- recordset: Record set ID assigned by iDigBio.
- rights
- tag
- type
- uuid: Unique identifier assigned by iDigBio.
- version: Media record version assigned by iDigBio.
- webstatement
- xpixels: As defined by EXIF, x dimension in pixel.
- ypixels: As defined by EXIF,y dimension in pixels.

Author(s)

Matthew Collins

Examples

```
## Not run:
# Searching for media using a query on related specimen information - first
# 10 media records with image URIs related to a specimen in the genus Acer:
df <- idig_search_media(rq=list(genus="acer"),
                        mq=list("data.ac:accessURI"=list("type"="exists")),
                        fields=c("uuid","data.ac:accessURI"), limit=10)

## End(Not run)
```

`idig_search_records` *Searching of iDigBio records*

Description

Function to query the iDigBio API for specimen records

Usage

```
idig_search_records(
  rq,
  fields = FALSE,
  max_items = 1e+05,
  limit = 0,
  offset = 0,
  sort = FALSE,
  ...
)
```

Arguments

`rq`	iDigBio record query in nested list format
`fields`	vector of fields that will be contained in the data.frame, limited set returned by default, use "all" to get all indexed fields
`max_items`	maximum number of results allowed to be retrieved (fail -safe)
`limit`	maximum number of results returned
`offset`	number of results to skip before returning results
`sort`	vector of fields to use for sorting, UUID is always appended to make paging safe
`...`	additional parameters

Details

Wraps `idig_search` to provide defaults specific to searching specimen records. Using this function instead of `idig_search` directly is recommened.

Queries need to be specified as a nested list structure that will serialize to an iDigBio query object's JSON as expected by the iDigBio API: `https://github.com/iDigBio/idigbio-search-api/wiki/Query-Format`

As an example, the first sample query looks like this in JSON in the API documentation:

```
{
  "scientificname": {
    "type": "exists"
  },
  "family": "asteraceae"
}
```

To rewrite this in R for use as the rq parameter to `idig_search_records` or `idig_search_media`, it would look like this:

```
rq <- list("scientificname"=list("type"="exists"),
           "family"="asteraceae"
           )
```

An example of a more complex JSON query with nested structures:

```
{
  "geopoint": {
   "type": "geo_bounding_box",
   "top_left": {
     "lat": 19.23,
     "lon": -130
    },
    "bottom_right": {
      "lat": -45.1119,
      "lon": 179.99999
    }
   }
 }
```

To rewrite this in R for use as the rq parameter, use nested calls to the list() function:

```
rq <- list(geopoint=list(
                         type="geo_bounding_box",
                         top_left=list(lat=19.23, lon=-130),
                         bottom_right=list(lat=-45.1119, lon= 179.99999)
                        )
           )
```

See the Examples section below for more samples of simpler and more complex queries. Please refer to the API documentation for the full functionality availible in queries.

All matching results are returned up to the max_items cap (default 100,000). If more results are wanted, a higher max_items can be passed as an option. This API loads records 5,000 at a time using HTTP so performance with large sets of data is not very good. Expect result sets over 50,000 records to take tens of minutes. You can use the `idig_count_records` or `idig_count_media` functions to find out how many records a query will return; these are fast.

The iDigBio API will only return 5,000 records at a time but this function will automatically page through the results and return them all. Limit and offset are availible if manual paging of results is needed though the max_items cap still applies. The item count comes from the results header not the count of actual records in the limit/offset window.

Return is a data.frame containing the requested fields (or the default fields). The columns in the data frame are untyped and no factors are pre- built. Attribution and other metadata is attached to the dataframe in the data.frame's attributes. (I.e. `attributes(df)`)

Value

A data frame with fields requested or the following default fields:

- UUID: Unique identifier assigned by iDigBio.
- occurrenceID
- catalognumber
- family - may be reassigned by iDigBio
- genus - may be reassigned by iDigBio
- scientificname - may be reassigned by iDigBio
- country - may be modified by iDigBio
- stateprovince
- geopoint: Assigned by iDigBio.
- data.dwc:eventDate
- data.dwc:year
- data.dwc:month
- data.dwc:day
- datecollected: May be reassigned by iDigBio, see more here
- collector: Assigned by iDigBio.
- recordset: Assigned by iDigBio.

Author(s)

Matthew Collins

Examples

```
## Not run:
# Simple example of retriving records in a genus:
idig_search_records(rq=list(genus="acer"), limit=10)

# This complex query shows that booleans passed to the API are represented
# as strings in R, fields used in the query don't have to be returned, and
# the syntax for accessing raw data fields:
idig_search_records(rq=list("hasImage"="true", genus="acer"),
            fields=c("uuid", "data.dwc:verbatimLatitude"), limit=100)

# Searching inside a raw data field for a string, note that raw data fields
# are searched as full text, indexed fields are search with exact matches:

idig_search_records(rq=list("data.dwc:dynamicProperties"="parasite"),
            fields=c("uuid", "data.dwc:dynamicProperties"), limit=100)

# Retriving a data.frame for use with MaxEnt. Notice geopoint is expanded
# to two columns in the data.frame: gepoint.lat and geopoint.lon:
df <- idig_search_records(rq=list(genus="acer", geopoint=list(type="exists")),
          fields=c("uuid", "geopoint"), limit=10)
write.csv(df[c("uuid", "geopoint.lon", "geopoint.lat")],
          file="acer_occurrences.csv", row.names=FALSE)


## End(Not run)
```

`idig_top_media` *Top media endpoint*

Description

Top media records summaries.

Usage

```
idig_top_media(rq = FALSE, mq = FALSE, top_fields = FALSE, count = 0, ...)
```

Arguments

`rq`	iDigBio record query in nested list format
`mq`	iDigBio media query in nested list format
`top_fields`	vector of field names to summarize by
`count`	maximum number of results to return, capped at 1000
`...`	additional parameters

Details

Summarize the count of media records in iDigBio according to unique values in the fields passed. This operates similarly to a SELECT DISTINCT count(field_name) query in SQL. When multiple fields are passed, the summaries are nested eg fields=c("country", "genus") would result in counting the top 10 genera in each of the top 10 countries for a total of 100 counts.

Value

nested list of field values with counts of media records

Author(s)

Matthew Collins

`idig_top_records` *Top records endpoint*

Description

Top specimen records summaries.

Usage

```
idig_top_records(rq = FALSE, top_fields = FALSE, count = 0, ...)
```

Arguments

`rq`	iDigBio record query in nested list format
`top_fields`	vector of field names to summarize by
`count`	maximum number of results to return, capped at 1000
`...`	additional parameters

Details

Summarize the count of specimen records in iDigBio according to unique values in the fields passed. This operates similarly to a SELECT DISTINCT count(field_name) query in SQL. When multiple fields are passed, the summaries are nested eg fields=c("country", "genus") would result in counting the top 10 genera in each of the top 10 countries for a total of 100 counts.

Value

nested list of field values with counts of specimen records

Author(s)

Matthew Collins

`idig_url` *base URL*

Description

Return base URL for the API calls.

Usage

```
idig_url(dev = FALSE)
```

Arguments

`dev`	Should be the beta version of the API be used?

Details

Defaults to use beta URL. Not exported.

Value

string for the URL

Author(s)

Francois Michonneau

`idig_validate` *validate fields*

Description

Stub function for validating parameters.

Usage

```
idig_validate(inputs)
```

Arguments

`inputs`	list of inputs to validate

Details

Takes list of inputs named by validation rule eg: `number:[2, 3]` and returns a vector of strings with any validation errors. If the vector is 0 length, everything is valid. Not exported.

Value

boolean

Author(s)

Matthew Collins

`idig_version` *API version*

Description

Return the version number to use for the API calls.

Usage

```
idig_version(version = "v2")
```

Arguments

`version`	optional argument giving the version of the API to use

Details

The current default is "v2". Not exported.

Value

string for the version to use

Author(s)

Francois Michonneau

`idig_view_media` *view media endpoint*

Description

View individual media records.

Usage

```
idig_view_media(uuid, ...)
```

Arguments

`uuid`	uuid of media record
`...`	additional parameters

Details

View all information about a specific media record.

Value

nested list of data

Author(s)

Matthew Collins

`idig_view_records` *view specimen endpoint*

Description

View individual specimen records.

Usage

```
idig_view_records(uuid, ...)
```

Arguments

`uuid`	uuid of specimen record
`...`	additional parameters

Details

View all information about a specific specimen record.

Value

nested list of data

Author(s)

Matthew Collins

`ignore_unused_imports` *Stub function for passing import checks*

Description

Stub function for passing import checks

Usage

```
ignore_unused_imports()
```

`ridigbio` *Retrieve data from the iDigBio specimen data repository.*

Description

Retrieve data from the iDigBio specimen data repository.

About

ridigbio provides an interface to the iDigBio data API described here: `https://www.idigbio.org/wiki/index.php/IDigBio_API`. With this package you can retrieve specimen and media records from the iDigBio data repository. The iDigBio portal `https://portal.idigbio.org/` uses the same API so you should be able to retrieve the same information as shown in the portal.

iDigBio contains nearly 30 million data records on museum specimens held at United States institutions. It also holds nearly 5 million images of these specimens.

Getting Started

The main function is `idig_search_records` and reviewing its documenation first with `?idig_search_records` is recommended.

Limitations

This package does not yet provide an interface to the mapping or the download APIs.

Citing

To cite the ridigbio package in your work, please use the following format:

Michonneau F, Collins M, Chamberlain SA (2016). ridigbio: An interface to iDigBio's search API that allows downloading specimen records. R package version 0.3.8. https://github.com/iDigBio/ridigbio

Author(s)

Francois Michonneau <`francois.michonneau@gmail.com`>

Matthew Collins <`mcollins@acis.ufl.edu`>

Index