

Package ‘SP2000’

August 4, 2020

Type Package

Title Catalogue of Life Toolkit

Version 0.1.0

Date 2020-08-04

Maintainer Liuyong Ding <ly_ding@126.com>

Description A programmatic interface to <<http://sp2000.org.cn>>, re-written based on an accompanying 'Species 2000' API. Access tables describing catalogue of the Chinese known species of animals, plants, fungi, micro-organisms, and more. This package also supports access to catalogue of life global <<http://catalogueoflife.org>> and catalogue of life Taiwan <https://taibnet.sinica.edu.tw/home_eng.php>. The development of 'SP2000' package were supported by Biodiversity Survey and Assessment Project of the Ministry of Ecology and Environment, China <2019HJ2096001006>, Yunnan University's ``Double First Class" Project <C176240405> and Yunnan University's Research Innovation Fund for Graduate Students <2019227>.

Depends R (>= 3.0.0)

Imports jsonlite, tibble, pbmcapply, purrr, rlist, XML, xml2, DT

Suggests utils, testthat

License Artistic-2.0

URL <https://github.com/Otoliths/SP2000>

BugReports <https://github.com/Otoliths/SP2000/issues>

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Liuyong Ding [aut, cre] (<<https://orcid.org/0000-0002-5490-182X>>),
Minrui Huang [ctb],
Ke Yang [ctb],
Jun Wang [ctb] (<<https://orcid.org/0000-0003-2481-1409>>),
Juan Tao [ctb],
Chengzhi Ding [ctb] (<<https://orcid.org/0000-0001-5215-7374>>),
Daming He [ctb]

Repository CRAN

Date/Publication 2020-08-04 07:30:02 UTC

R topics documented:

SP2000-package	2
download_col_china	4
find_synonyms	5
get_col_global	7
get_col_taiwan	8
get_province	10
get_redlist_china	10
get_top20	12
list_df	13
open_url	14
search_checklist	15
search_family_id	16
search_taxon_id	17
set_search_key	19

Index 20

SP2000-package *Catalogue of Life Toolkit*

Description

This package is designed for mining the checklist of animals, plants, fungi and micro-organisms both in and outside China.

Details

Package: SP2000
 Type: Package
 Title: Catalogue of Life Toolkit
 Version: 0.1.0
 Date: 2020-08-04
 Authors@R: c(person(given = "Liuyong", family = "Ding", email = "ly_ding@126.com", role = c("aut", "cre"), comment = "")
 Maintainer: Liuyong Ding <ly_ding@126.com>
 Description: A programmatic interface to <http://sp2000.org.cn>, re-written based on an accompanying 'Species 2000' A
 Depends: R (>= 3.0.0)
 Imports: jsonlite, tibble, pbmcapply, purrr, rlist, XML, xml2, DT
 Suggests: utils, testthat
 License: Artistic-2.0
 URL: <https://github.com/Otoliths/SP2000>
 BugReports: <https://github.com/Otoliths/SP2000/issues>

Encoding: UTF-8
 LazyData: true
 RoxygenNote: 7.1.1
 Author: Liuyong Ding [aut, cre] (<<https://orcid.org/0000-0002-5490-182X>>), Minrui Huang [ctb], Ke Yang [ctb], Jun

Index of help topics:

SP2000-package	Catalogue of Life Toolkit
download_col_china	Download 'Catalogue of Life China': Annual Checklist
find_synonyms	Find synonyms via species name
get_col_global	Search Catalogue of Life Global checklist
get_col_taiwan	Search Catalogue of Life Taiwan checklist
get_province	Statistics on Species or Intraspecies by Province from Catalogue of Life China checklist
get_redlist_china	Query Redlist of Chinese Biodiversity
get_top20	Download the top20 species for Catalogue of Life China checklist
list_df	Catalogue of Life list(s) convert data frame
open_url	Opening the Catalogue of Life China checklist web pages with query
search_checklist	Search Catalogue of Life China checklist
search_family_id	Search family IDs
search_taxon_id	Search taxon IDs
set_search_key	SP2000 API keys

Author(s)

Maintainer: Liuyong Ding <ly_ding@126.com>

References

Ding LY, Li H, Tao J, Zhang JL, Huang MR, Yang K, Wang J, He DM, Ding CZ (2020) SP2000: An open source R package for querying the catalogue of life. Biodiversity Science.

<https://cran.r-project.org/package=SP2000>

<https://pypi.org/project/SP2000>

Examples

```
## Not run:
# Note: You need to apply for the apiKey <http://sp2000.org.cn/api/document>
to run search_* functions of this package.
```

```
## Load "SP2000"
```

```
library('SP2000')
```

```
## Set your key
```

```

set_search_key <- "your apiKey"

## Search family IDs via family name, supports Latin and Chinese names

familyid <- search_family_id(query = "Anguillidae")

## Search taxon IDs via familyID ,scientificName and commonName

query <- familyid$Anguillidae$data$record_id

taxonid <- search_taxon_id(query = query,name = "familyID")

queries = c("Anguilla marmorata","Anguilla japonica",
            "Anguilla bicolor","Anguilla nebulosa",
            "Anguilla luzonensis")

search_taxon_id(query = queries,name = "scientificName")

## Download detailed lists via species or infraspecies ID

query <- taxonid[["3851c5311bed46c19529cb155d37aa9b"]][["data"]][["namecode"]]

checklist <- search_checklist(query = query)

## Get Catalogue of Life Global checklist via species name and id

x <- get_col_global(query = "Anguilla", response = "full")

str(x)

x[["Anguilla"]][["meta"]][["total_number_of_results"]] [1]

## Find synonyms via species name from Catalogue of Life Global

find_synonyms(query = queries)

## Search Catalogue of Life Taiwan checklist

get_col_taiwan(query = "Anguillidae", level = "family")

## Query Redlist of Chinese Biodiversity

get_redlist_china(query = "Anguilla", option = "Scientific Names")

## End(Not run)

```

Description

Organized by the Biodiversity Committee of Chinese Academy of Sciences (BC-CAS), Catalogue of Life China Annual Checklist edition has been compiled by Species 2000 China Node.

Usage

```
download_col_china(version = "2020", OS = "MacOS", dir = tempdir())
```

Arguments

version	integer Release version of annual checklist,the default value is 2020.
OS	character Supported operating system,c("MacOS", "Ubuntu", "Windows"),the default value is "MacOS".
dir	a non-empty character vector giving the directory name by user,the default value is dir = tempdir(),see tempdir for details.

Details

Visit the website <http://sp2000.org.cn/download> for more details.

Value

URL

Author(s)

Liuyong Ding <ly_ding@126.com>

Examples

```
dir <- tempdir()
download_col_china(version = "2020",OS = "MacOS", dir = dir)
```

find_synonyms	<i>Find synonyms via species name</i>
---------------	---------------------------------------

Description

Find synonyms via species name from Catalogue of Life Global.

Usage

```
find_synonyms(query, mc.cores = 2)
```

Arguments

query	character species name,The function is similar to get_col_global .
mc.cores	The number of cores to use, i.e. at most how many child processes will be run simultaneously. The option is initialized from environment variable MC_CORES if set. Must be at least one, and parallelization requires at least two cores,see mclapply for details.

Details

Visit the website <http://webservice.catalogueoflife.org/col/webservice> for more details.

Value

object

Author(s)

Liuyong Ding <ly_ding@126.com>

References

https://github.com/lutteropp/SpeciesSynonymFinder/blob/master/find_synonyms.r

Examples

```
## Not run:
##Get Catalogue of Life Global checklist via species name
x1 <- get_col_global(query = c("Anguilla marmorata","Anguilla japonica",
                             "Anguilla bicolor","Anguilla nebulosa",
                             "Anguilla luzonensis"),
                   option = "name")

str(x1)

##full queries
x2 <- get_col_global(query = "Anguilla", response = "full")

##Find synonyms via species name
find_synonyms(query = c("Anguilla marmorata","Anguilla japonica",
                       "Anguilla bicolor","Anguilla nebulosa",
                       "Anguilla luzonensis"))

## End(Not run)
```

 get_col_global

 Search Catalogue of Life Global checklist

Description

Get Catalogue of Life Global checklist via species name and id.

Usage

```
get_col_global(
  query,
  option = "name",
  response = "terse",
  start = 0,
  limit = 500,
  mc.cores = 2
)
```

Arguments

query	string	The string to search for.
option	character	There is one required parameter, which is either name or id. Give either a name or an ID. If an ID is given, the name parameter may not be used, and vice versa. option=c("id","name"),the default value is "name". Only exact matches found the name given will be returned, unless a wildcard (*) is appended. Wildcards are allowed only at the end of the string. This offers the option to e.g. search for genus* to retrieve the genus plus all its (infra)species. The name must be at least 3 characters long, not counting the wildcard character. The record ID of the specific record to return (only for scientific names of species or infraspecific taxa).
response	character	Type of response returned. Valid values are response=terse and response=full. if the response parameter is omitted, the results are returned in the default terse format. If format=terse then a minimum set of results are returned (this is faster and smaller, enough for name lookup), if format=full then all available information is returned, response=c("full","terse"),the default value is "terse".
start	integer	Record number to start at. If omitted, the results are returned from the first record (start=0). Use in combination with limit to page through results. Note that we do the paging internally for you, but you can manually set the start parameter.
limit	integer	Number of records to return. This is useful if the total number of results is larger than the maximum number of results returned by a single Web service query (currently the maximum number of results returned by a single query is 500 for terse queries and 50 for full queries,the default value is 500.Note that there is a hard maximum of 10,000, which is calculated as the limit+start, so start=99,00 and limit=2000 won't work.

mc.cores The number of cores to use, i.e. at most how many child processes will be run simultaneously. The option is initialized from environment variable MC_CORES if set. Must be at least one, and parallelization requires at least two cores, see [mclapply](#) for details.

Details

Visit the website <http://webservice.catalogueoflife.org/col/webservice> for more details.

Value

object

Author(s)

Liuyong Ding <ly_ding@126.com>

Examples

```
## Not run:
##Get Catalogue of Life Global checklist via species name
x1 <- get_col_global(query = c("Anguilla marmorata", "Anguilla japonica",
                             "Anguilla bicolor", "Anguilla nebulosa",
                             "Anguilla luzonensis"),
                   option = "name")

str(x1)

##full queries
x2 <- get_col_global(query = "Anguilla", response = "full")

##Find synonyms via species name
find_synonyms(query = c("Anguilla marmorata", "Anguilla japonica",
                       "Anguilla bicolor", "Anguilla nebulosa",
                       "Anguilla luzonensis"))

## End(Not run)
```

get_col_taiwan

Search Catalogue of Life Taiwan checklist

Description

Get Catalogue of Life Taiwan checklist via advanced query.

Usage

```
get_col_taiwan(  
  query,  
  level = "species",  
  option = "equal",  
  include_synonyms = TRUE  
)
```

Arguments

query	string	The string to search for.
level	character	Query by category level, level=c("kingdom","phylum","class","order","family","genus","species"), default value is "species".
option	character	Query format, option=c("contain","equal","beginning"),the default value is "equal".
include_synonyms	logic	Whether the results contain a synonym or not.

Details

Visit the website https://taibnet.sinica.edu.tw/eng/taibnet_species_query.php for more details.

Value

object

Author(s)

Liuyong Ding <ly_ding@126.com>

Examples

```
## Not run:  
get_col_taiwan(query="Anguilla",level="species",option = "contain")  
  
get_col_taiwan(query="Anguillidae",level="family")  
  
## End(Not run)
```

get_province	<i>Statistics on Species or Intraspecies by Province from Catalogue of Life China checklist</i>
--------------	---

Description

Download the statistics datasets on species/intraspecies by province at http://sp2000.org.cn/statistics/statistics_map for more details.

Usage

```
get_province()
```

Details

Visit the website http://sp2000.org.cn/statistics/statistics_map for more details.

Value

Statistics on species or intraspecies by province.

Author(s)

Liuyong Ding

Liuyong Ding <ly_ding@126.com>

Examples

```
get_province()
```

get_redlist_china	<i>Query Redlist of Chinese Biodiversity</i>
-------------------	--

Description

Query Redlist of China's Biodiversity of Vertebrate, Higher Plants and Macrofungi.

Usage

```
get_redlist_china(  
  query = NULL,  
  option = "Scientific Names",  
  group = "Amphibians",  
  viewDT = FALSE  
)
```

Arguments

query	string	The string to query for.
option	character	There is one required parameter, which is either Chinese Names or Scientific Names. Give either a Chinese Names or Scientific Names. If an Scientific Names is given, the Chinese Names parameter may not be used. Only exact matches found the name given will be returned. option=c("Chinese Names","Scientific Names"),the default value is "Scientific Names".
group	character	There is one required parameter, group=c("Amphibians","Birds","Inland Fishes","Mammals","Reptiles","Plants","Fungi").
viewDT	logic	TRUE or FALSE,the default value is FALSE.

Format

assessment status:

EX Extinct

EW Extinct in the wild

RE Regional Extinct

CR Critically Endangered

EN Endangered

VU Vulnerable

NT Near Threatened

LC Least Concern

DD Data Deficient

Details

Visit the website <http://www.mee.gov.cn> for more details.

Value

object

Author(s)

Liuyong Ding <ly_ding@126.com>

Ke Yang <ydyangke@163.com>

References

<http://www.mee.gov.cn>

http://www.mee.gov.cn/xxgk2018/xxgk/xxgk01/201805/t20180524_629586.html

http://www.mee.gov.cn/gkml/hbb/bgg/201309/t20130912_260061.html

http://www.mee.gov.cn/gkml/hbb/bgg/201505/t20150525_302233.html

Examples

```
#query assessment status via Chinese Names or Scientific Names
get_redlist_china(query = "Anguilla", option = "Scientific Names")
get_redlist_china(query = "Anguilla nebulosa", option = "Scientific Names")

#creates an HTML widget to display rectangular data
get_redlist_china(group = "Inland Fishes", viewDT = TRUE)
```

get_top20

Download the top20 species for Catalogue of Life China checklist

Description

Download the most visited top20 species from <http://sp2000.org.cn> for more details.

Usage

```
get_top20()
```

Details

Visit the website <http://sp2000.org.cn> for more details.

Value

top20 species

Author(s)

Liuyong Ding

Examples

```
get_top20()
```

list_df	<i>Catalogue of Life list(s) convert data frame</i>
---------	---

Description

Checklist lists convert data frame.

Usage

```
list_df(x, db = c("colchina", "colglobal"))
```

Arguments

x	list Results returned by the function search_checklist and get_col_global .
db	character db = c("colchina","colglobal")

Format

A data frame with 19 variables:

ScientificName The scientific name (the accepted name) includes the name and the date of the name

Synonyms Synonyms name, Latin

ChineseName Chinese name

CommonNames Common name

Kingdom Kingdom at taxonTree

Phylum Phylum at taxonTree

Class Class at taxonTree

Order Order at taxonTree

Family Family at taxonTree

Genus Genus at taxonTree

Species Species at taxonTree

Infraspecies Infraspecies at taxonTree

Distribution Distribution of species or infraspecies

Name Full name of reviewer in English or Chinese

Email Organization of the reviewer in English or Chinese

Address Email address of the reviewer

Institution Address of the reviewer in English or Chinese

References References

Download Download date

Author(s)

Liuyong Ding <ly_ding@126.com>

Source

Visit the website <http://sp2000.org.cn/api/document> for more details

Examples

```
## Not run:
##Set your key
set_search_key <- "your apiKey"

##Search family IDs via family name
familyid <- search_family_id(query = "Anguillidae")

##Search taxon IDs via familyID
taxonid <- search_taxon_id(query = familyid$Anguillidae$data$record_id, name = "familyID")

#Download detailed lists via species or infraspecies ID
query <- taxonid[["3851c5311bed46c19529cb155d37aa9b"]][["data"]][["namecode"]]
x1 <- search_checklist(query = query)
str(x1)
x1 <- list_df(x1,db = "colchina")

#Get Catalogue of Life Global checklist via species name
x2 <- get_col_global(query = c("Anguilla marmorata","Anguilla japonica",
                             "Anguilla bicolor","Anguilla nebulosa",
                             "Anguilla luzonensis"),
                   option = "name")

str(x2)
x2 <- list_df(x2,db = "colglobal")

## End(Not run)
```

open_url

Opening the Catalogue of Life China checklist web pages with query

Description

Query with the acceptedName,scientificName,chineseName and commonName, returning the web pages that meets the criteria.

Usage

```
open_url(query = NULL, name = "acceptedName", language = "en")
```

Arguments

query	string the acceptedName,scientificName,chineseName or commonName.
name	string name = c("acceptedName","scientificName","chineseName","commonName"),the default value is "acceptedName".
language	string currently only two languages are supported: Chinese and English,language=c("en","zh"), the default value is "en".

Details

Visit the website <http://sp2000.org.cn/pageservices/document> for more details.

Author(s)

Liuyong Ding

Liuyong Ding <ly_ding@126.com>

Examples

```
open_url(query = "Anguilla marmorata",name = "scientificName",language = 'en')
open_url(query = "Anguilla marmorata",name = "scientificName",language = 'zh')
```

search_checklist	<i>Search Catalogue of Life China checklist</i>
------------------	---

Description

Get checklist via species or infraspecies ID.

Usage

```
search_checklist(query = NULL, mc.cores = 2)
```

Arguments

query	string One or more queries, see search_family_id and search_taxon_id for more details.
mc.cores	The number of cores to use, i.e. at most how many child processes will be run simultaneously. The option is initialized from environment variable MC_CORES if set. Must be at least one, and parallelization requires at least two cores,see mclapply for details.

Details

Visit the website <http://sp2000.org.cn/api/document> for more details.

Value

Catalogue of Life China list(s)

Author(s)

Liuyong Ding <ly_ding@126.com>

Examples

```
## Not run:
##Set your key
set_search_key <- "your apiKey"

##Search family IDs via family name
familyid <- search_family_id(query = "Anguillidae")

##Search taxon IDs via familyID
taxonid <- search_taxon_id(query = familyid$Anguillidae$data$record_id,name = "familyID")

#Download detailed lists via species or infraspecies ID
query <- taxonid[["3851c5311bed46c19529cb155d37aa9b"]][["data"]][["namecode"]]
x <- search_checklist(query = query)
str(x)

## End(Not run)
```

search_family_id	<i>Search family IDs</i>
------------------	--------------------------

Description

Search family IDs via family name, supports Latin and Chinese names.

Usage

```
search_family_id(query = NULL, start = 1, limit = 20, mc.cores = 2)
```

Arguments

query	character One and more queries,support Family name, or part of family name, supports Latin and Chinese names.
start	integer Record number to start at. If omitted, the results are returned from the first record (start=1). Use in combination with limit to page through results. Note that we do the paging internally for you, but you can manually set the start parameter.
limit	integer Number of records to return, the default value is 20.

mc.cores The number of cores to use, i.e. at most how many child processes will be run simultaneously. The option is initialized from environment variable MC_CORES if set. Must be at least one, and parallelization requires at least two cores, see [mclapply](#) for details.

Details

Visit the website <http://sp2000.org.cn/api/document> for more details

Value

dataframe

Author(s)

Liuyong Ding <ly_ding@126.com>

Examples

```
## Not run:
##Set your key
set_search_key <- "your apiKey"

##Search family IDs via family name
familyid <- search_family_id(query = "Anguillidae")

## End(Not run)
```

search_taxon_id	<i>Search taxon IDs</i>
-----------------	-------------------------

Description

Search taxon IDs via familyID ,scientificName and commonName.

Usage

```
search_taxon_id(
  query = NULL,
  name = "scientificName",
  start = 1,
  limit = 20,
  mc.cores = 2
)
```

Arguments

query	string familyID ,scientificName or commonName.
name	character name = c("familyID","scientificName","commonName"),the default value is "scientificName".
start	intenger Record number to start at. If omitted, the results are returned from the first record (start=1). Use in combination with limit to page through results. Note that we do the paging internally for you, but you can manually set the start parameter.
limit	intenger Number of records to return. This is passed across all sources,when you first query, set the limit to something smallish so that you can get a result quickly, then do more as needed.
mc.cores	The number of cores to use, i.e. at most how many child processes will be run simultaneously. The option is initialized from environment variable MC_CORES if set. Must be at least one, and parallelization requires at least two cores,see mclapply for details.

Format

query:

taxonIDs an array of species' ids

familyID family ID, unique value

scientificName the scientific name, or part of the scientific name, supports Latin names and Chinese

commonName common name, or part of common name

Details

Visit the website <http://sp2000.org.cn/api/document> for more details

Value

dataframe

Author(s)

Liuyong Ding <ly_ding@126.com>

Examples

```
## Not run:
##Set your key
set_search_key <- "your apiKey"

##Search family IDs via family name
familyid <- search_family_id(query = "Anguillidae")

##Search taxon IDs via familyID
```

```
taxonid <- search_taxon_id(query = familyid$Anguillidae$data$record_id,name = "familyID")

## End(Not run)
```

set_search_key	<i>SP2000 API keys</i>
----------------	------------------------

Description

Apply for the apiKey variable to be used by all search_* functions, register for <http://sp2000.org.cn/api/document> and use an API key. This function allows users to set this key. Note: The daily API visits of ordinary users are 2000, If you want to apply for increasing the daily API request limit, please fill in the application form <http://col.especies.cn/doc/API.docx> and send an email to <SP2000CN@ibcas.ac.cn> entitled "Application for increasing API Request Times".

Usage

```
set_search_key(key)
```

Arguments

key string Value to set apiKey to (i.e. your API key).

Value

A logical of length one, TRUE is the value was set FALSE if not. value is returned inside invisible(), i.e. it is not printed to screen when the function is called.

Author(s)

Liuyong Ding <ly_ding@126.com>

Examples

```
## Not run:
#Set the apiKey variable to be used by all search_* functions
set_search_key("your apiKey")

## End(Not run)
```

Index

* **package**

SP2000-package, [2](#)

download_col_china, [4](#)

find_synonyms, [5](#)

get_col_global, [6](#), [7](#), [13](#)

get_col_taiwan, [8](#)

get_province, [10](#)

get_redlist_china, [10](#)

get_top20, [12](#)

list_df, [13](#)

mclapply, [6](#), [8](#), [15](#), [17](#), [18](#)

open_url, [14](#)

search_checklist, [13](#), [15](#)

search_family_id, [15](#), [16](#)

search_taxon_id, [15](#), [17](#)

set_search_key, [19](#)

SP2000 (SP2000-package), [2](#)

SP2000-package, [2](#)

tempdir, [5](#)