Package 'rfishbase'

June 3, 2023

Title R Interface to 'FishBase'

Description A programmatic interface to 'FishBase', re-written based on an accompanying 'RESTful' API. Access tables describing over 30,000 species of fish, their biology, ecology, morphology, and more. This package also supports experimental access to 'SeaLifeBase' data, which contains nearly 200,000 species records for all types of aquatic life not covered by 'FishBase.'

Version 4.1.2

Encoding UTF-8

License CC0

URL https://docs.ropensci.org/rfishbase/, https://github.com/ropensci/rfishbase

BugReports https://github.com/ropensci/rfishbase/issues

LazyData true

Depends R (>= 4.0)

Imports methods, utils, tools, purrr, progress, memoise, rlang, magrittr, readr (>= 2.0.0), stringr, jsonlite, DBI, dplyr, dbplyr, duckdb, contentid (>= 0.0.15), rstudioapi, fs, glue, tibble

Suggests testthat, rmarkdown, knitr, covr, spelling

RoxygenNote 7.2.3

Language en-US

NeedsCompilation no

Author Carl Boettiger [cre, aut] (<https://orcid.org/0000-0002-1642-628X>), Scott Chamberlain [aut] (<https://orcid.org/0000-0003-1444-9135>), Duncan Temple Lang [aut], Peter Wainwright [aut], Kevin Cazelles [ctb] (<https://orcid.org/0000-0001-6619-9874>)

Maintainer Carl Boettiger <cboettig@gmail.com>

Repository CRAN

Date/Publication 2023-06-02 22:20:02 UTC

R topics documented:

`rfishbase-package` *The new R interface to Fishbase, v2.0*

Description

A programmatic interface to FishBase, re-written based on an accompanying 'RESTful' API. Access tables describing over 30,000 species of fish, their biology, ecology, morphology, and more. This package also supports experimental access to SeaLifeBase data, which contains nearly 200,000 species records for all types of aquatic life not covered by FishBase.'

Author(s)

Carl Boettiger <carl@ropensci.org>

Scott Chamberlain <scott@ropensci.org>

`available_releases` *List available releases*

Description

List available releases

Usage

```
available_releases(server = c("fishbase", "sealifebase"))
```

Arguments

`server`	fishbase or sealifebase

Details

Lists all available releases (year.month format). To use a specific release, set the desired release using ‘options(FISHBASE_VERSION=)‘, as shown in the examples. Otherwise, rfishbase will use the latest available version if this option is unset. NOTE: it will be necessary to clear the cache with ‘clear_cache()‘ or by restarting the R session with a fresh environment.

Examples

```
available_releases()
options(FISHBASE_VERSION="19.04")
## unset
options(FISHBASE_VERSION=NULL)
```

`brains` *brains*

Description

brains

Usage

```
brains(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

`species_list`	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
`fields`	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable ‘FISHBASE_API‘, e.g. ‘Sys.setenv(FISHBASE_API="sealifebase")‘.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
`db`	the
`...`	unused; for backwards compatibility only

Value

a table of species brains

Examples

```
 ## Not run:
brains("Oreochromis niloticus")

## End(Not run)
```

`common_names` *common names*

Description

Return a table of common names

Usage

```
common_names(
  species_list = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(),
  Language = "English",
  fields = NULL
)
```

Arguments

`species_list`	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
`db`	the
`Language`	a string specifying the language for the common name, e.g. "English"
`fields`	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later

Details

Note that there are many common names for a given sci name, so sci_to_common doesn't make sense

Value

a data.frame of common names by species queried. If multiple species are queried, The resulting data.frames are concatenated.

`common_to_sci` *common_to_sci*

Description

Return a list of scientific names corresponding to given the common name(s).

Usage

```
common_to_sci(
  x,
  Language = "English",
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db()
)
```

Arguments

`x`	a common name or list of common names
`Language`	a string specifying the language for the common name, e.g. "English"
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
`db`	the

Details

If more than one scientific name matches the common name (e.g. "trout"), the function will simply return a list of all matching scientific names. If given more than one common name, the resulting strings of matching scientific names are simply concatenated.

Value

a character vector of scientific names

See Also

species_list, synonyms

Examples

```
common_to_sci(c("Bicolor cleaner wrasse", "humphead parrotfish"), Language="English")
common_to_sci(c("Coho Salmon", "trout"))
```

---

country *country*

Description

return a table of country for the requested species, as reported in FishBASE.org

Usage

```
country(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Details

e.g. http://www.fishbase.us/Country

Examples

```
## Not run:
country(species_list(Genus='Labroides'))

## End(Not run)
```

`countrysub` *countrysub*

Description

return a table of countrysub for the requested species

Usage

```
countrysub(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

`species_list`	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
`fields`	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
`db`	the
`...`	unused; for backwards compatibility only

Examples

```
## Not run:
countrysub(species_list(Genus='Labroides'))

## End(Not run)
```

---

countrysubref *countrysubref*

Description

return a table of countrysubref

Usage

```
countrysubref(
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(),
  ...
)
```

Arguments

server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Examples

```
## Not run:
countrysubref()

## End(Not run)
```

c_code *c_code*

Description

return a table of country information for the requested c_code, as reported in FishBASE.org

Usage

```
c_code(
  c_code = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(),
  ...
)
```

Arguments

c_code	a C_Code or list of C_Codes (FishBase country code)
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Details

e.g. http://www.fishbase.us/Country

Examples

```
## Not run:
c_code(440)

## End(Not run)
```

db_dir *show fishbase directory*

Description

show fishbase directory

Usage

```
db_dir()
```

db_disconnect *disconnect the database*

Description

disconnect the database

Usage

```
db_disconnect(db = NULL)
```

Arguments

db	optional, an existing pointer to the db, e.g. from [fb_conn()] or [fb_import()].

diet *diet*

Description

diet

Usage

```
diet(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species diet

References

http://www.fishbase.org/manual/english/fishbasethe_diet_table.htm

Examples

```
## Not run:
diet()

## End(Not run)
```

---

diet_items *diet_items*

Description

diet_items

Usage

```
diet_items(...)
```

Arguments

...	additional arguments (not used)

Value

a table of diet_items

Examples

```
## Not run:
diet_items()

## End(Not run)
```

`distribution` *distribution*

Description

return a table of species locations as reported in FishBASE.org FAO location data

Usage

```
distribution(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(),
  ...
)
```

Arguments

`species_list`	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
`fields`	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
`db`	the
`...`	unused; for backwards compatibility only

Details

currently this is ~ FAO areas table (minus "note" field) e.g. http://www.fishbase.us/Country/FaoAreaList.php?ID=5537

Examples

```
## Not run:
distribution(species_list(Genus='Labroides'))

## End(Not run)
```

---

docs *docs*

Description

documentation of tables and fields

Usage

```
docs(table = NULL, server = NULL, ...)
```

Arguments

table	the table for which the documentation should be displayed. If no table is given, documentation summarizing all available tables is shown.
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
...	unused; for backwards compatibility only

Value

A data.frame which lists the name of each table (if no table argument is given), along with a description of the table and a URL linking to further information about the table. If a specific table is named in the table argument, then the function will return a data.frame listing all the fields (columns) found in that table, a description of what the field label means, and the units in which the field is measured. These descriptions of the columns are not made available by FishBase and must be manually generated and curated by FishBase users. At this time, many fields are still missing. Please take a moment to fill in any fields you use in the source table here: https://github.com/ropensci/fishbaseapi/tree/master/docs/docs-sources

Examples

```
tables <- docs()
# Describe the fecundity table
dplyr::filter(tables, table == "fecundity")$description
## See fields in fecundity table
docs("fecundity")
## Note: only
```

---

ecology *ecology*

Description

ecology

Usage

```
ecology(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Details

By default, will only return one entry (row) per species. Increase limit to get multiple returns for different stocks of the same species, though often data is either identical to the first or simply missing in the additional stocks.

Value

a table of species ecology data

References

http://www.fishbase.org/manual/english/fishbasethe_ecology_table.htm

Examples

```
## Not run:
ecology("Oreochromis niloticus")

## trophic levels and standard errors for a list of species
ecology(c("Oreochromis niloticus", "Salmo trutta"),
        fields=c("SpecCode", "FoodTroph", "FoodSeTroph", "DietTroph", "DietSeTroph"))

## End(Not run)
```

`ecosystem` *ecosystem*

Description

ecosystem

Usage

```
ecosystem(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species ecosystems data

Examples

```
## Not run:
ecosystem("Oreochromis niloticus")

## End(Not run)
```

---

estimate *estimate*

Description

estimate

Usage

```
estimate(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of estimates from some models on trophic levels

References

http://www.fishbase.us/manual/English/FishbaseThe_FOOD_ITEMS_table.htm

Examples

```
## Not run:
estimate("Oreochromis niloticus")

## End(Not run)
```

---

faoareas *faoareas*

Description

return a table of species locations as reported in FishBASE.org FAO location data

Usage

```
faoareas(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Details

currently this is ~ FAO areas table (minus "note" field) e.g. http://www.fishbase.us/Country/FaoAreaList.php?ID=5537

Value

a tibble, empty tibble if no results found

Examples

```
## Not run:
  faoareas()

## End(Not run)
```

---

fb_conn *Cacheable database connection*

Description

Cacheable database connection

Usage

```
fb_conn(server = c("fishbase", "sealifebase"), version = "latest")
```

Arguments

server	fishbase or sealifebase
version	release version

fb_import *Import tables to local store*

Description

Import tables to local store

Usage

```
fb_import(
  server = c("fishbase", "sealifebase"),
  version = get_latest_release(),
  db = fb_conn(server, version),
  tables = NULL
)
```

Arguments

server	fishbase or sealifebase
version	release version
db	A cachable duckdb database connection
tables	list of tables to import. Default 'NULL' will import all tables.

Details

Downloads and stores tables from the requested version of fishbase or sealifebase. If the table is already downloaded, it will not be re-downloaded. Imported tables are added to the active duckdb connection. Note that there is no need to call this

Examples

```
conn <- fb_import()
```

fb_tables *fb_tables list tables*

Description

fb_tables list tables

Usage

```
fb_tables(server = c("fishbase", "sealifebase"), version = "latest")
```

Arguments

server	fishbase or sealifebase
version	release version

`fb_tbl` *Access a fishbase or sealifebase table*

Description

Access a fishbase or sealifebase table

Usage

```
fb_tbl(
  tbl,
  server = c("fishbase", "sealifebase"),
  version = "latest",
  db = fb_conn(server, version),
  collect = TRUE
)
```

Arguments

tbl	table name, as it appears in the database. See [fb_tables()] for a list.
server	fishbase or sealifebase
version	release version
db	A cachable duckdb database connection
collect	should we return an in-memory table? Generally best to leave as TRUE unless RAM is too limited. A remote table can be used with most dplyr functions (filter, select, joins, etc) to further refine.

Examples

```
fb_tbl("species")
```

fecundity *fecundity*

Description

fecundity

Usage

```
fecundity(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species fecundity

Examples

```
## Not run:
fecundity("Oreochromis niloticus")

## End(Not run)
```

fishbase *A table of all the the species found in FishBase, including taxonomic classification and the Species Code (SpecCode) by which the species is identified in FishBase.*

Description

A table of all the the species found in FishBase, including taxonomic classification and the Species Code (SpecCode) by which the species is identified in FishBase.

Author(s)

Carl Boettiger <carl@ropensci.org>

fishbase_pane *Open database connection pane in RStudio*

Description

This function launches the RStudio "Connection" pane to interactively explore the database.

Usage

```
fishbase_pane()
```

Examples

```
if (!is.null(getOption("connectionObserver"))) fishbase_pane()
```

fooditems *fooditems*

Description

fooditems

Usage

```
fooditems(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species fooditems

References

http://www.fishbase.org/manual/english/fishbasethe_food_items_table.htm

Examples

```
## Not run:
fooditems("Oreochromis niloticus")

## End(Not run)
```

genetics *genetics*

Description

genetics

Usage

```
genetics(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species genetics data

Examples

```
## Not run:
genetics("Oreochromis niloticus")
genetics("Labroides dimidiatus")

## End(Not run)
```

`introductions` *introductions*

Description

introductions

Usage

```
introductions(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

`species_list`	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
`fields`	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
`...`	unused; for backwards compatibility only

Value

a table of species introductions data

Examples

```
## Not run:
introductions("Oreochromis niloticus")

## End(Not run)
```

larvae *larvae*

Description

larvae

Usage

```
larvae(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of larval data

Examples

```
## Not run:
larvae("Oreochromis niloticus")

## End(Not run)
```

length_freq *length_freq*

Description

return a table of species fooditems

Usage

```
length_freq(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of length_freq information by species; see details

References

http://www.fishbase.org/manual/english/lengthfrequency.htm

Examples

```
## Not run:
length_freq("Oreochromis niloticus")

## End(Not run)
```

`length_length` *length_length*

Description

return a table of lengths

Usage

```
length_length(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

`species_list`	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
`fields`	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
`db`	the
`...`	unused; for backwards compatibility only

Details

This table contains relationships for the conversion of one length type to another for over 8,000 species of fish, derived from different publications, e.g. Moutopoulos and Stergiou (2002) and Gaygusuz et al (2006), or from fish pictures, e.g. Collette and Nauen (1983), Compagno (1984) and Randall (1997). The relationships, which always refer to centimeters, may consist either of a regression linking two length types, of the form: Length type (2) = a + b x Length type (1) Length type (2) = b' x Length type (1) The available length types are, as elsewhere in FishBase, TL = total length; FL = fork length; SL = standard length; WD = width (in rays); OT = other type (to be specified in the Comment field). When a version of equation (1) is presented, the length range, the number of fish used in the regression, the sex and the correlation coefficient are presented, if available. When a version of equation (2) is presented, the range and the correlation coefficient are omitted, as the ratio in (2) will usually be estimated from a single specimen, or a few fish covering a narrow range of lengths.

Value

a table of lengths

References

http://www.fishbase.org/manual/english/PDF/FB_Book_CBinohlan_Length-Length_RF_JG.pdf

Examples

```
## Not run:
length_length("Oreochromis niloticus")

## End(Not run)
```

`length_weight` *length_weight*

Description

The LENGTH-WEIGHT table presents the a and b values of over 5,000 length-weight relationships of the form W = a x Lb, pertaining to about over 2,000 fish species.

Usage

```
length_weight(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Details

See references for official documentation. From FishBase.org: Length-weight relationships are important in fisheries science, notably to raise length-frequency samples to total catch, or to estimate biomass from underwater length observations. The units of length and weight in FishBase are centimeter and gram, respectively. Thus when length-weight relationships are not in cm-g, the intercept 'a' is transformed as follows:

a'(cm, g) = a (mm, g)*10^b a'(cm, g) = a (cm, kg)*1000 a'(cm, g) = a (mm, mg)*10^b/1000 a'(cm, g) = a (mm, kg)*10^b*1000

However, published length-weight relationships are sometimes difficult to use, as they may be based on a length measurement type (e.g., fork length) different from ones length measurements (expressed e.g., as total length). Therefore, to facilitate conversion between length types, an additional LENGTH-LENGTH table, #' presented below, was devised which presents linear regressions or ratios linking length types (e.g., FL vs. TL). We included a calculated field with the weight of a 10 cm fish (which should be in the order of 10 g for normal, fusiform shaped fish), to allow identification of gross errors, given knowledge of the body form of a species.

Value

a table of length_weight information by species; see details

References

http://www.fishbase.org/manual/english/fishbasethe_length_weight_table.htm

Examples

```
## Not run:
length_weight("Oreochromis niloticus")

## End(Not run)
```

load_taxa *load_taxa*

Description

load_taxa

Usage

```
load_taxa(
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  collect = TRUE,
  ...
)
```

Arguments

server	Either "fishbase" (the default) or "sealifebase"
version	the version of the database you want. Will default to the latest avialable; see [available_releases()].
db	A remote database connection. Will default to the best available system, see [default_db()].
collect	return a data.frame if TRUE, otherwise, a DBI connection to the table in the database
...	for compatibility with previous versions

Value

the taxa list

maturity *maturity*

Description

maturity

Usage

```
maturity(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species maturity

Examples

```
## Not run:
maturity("Oreochromis niloticus")

## End(Not run)
```

`morphology` *morphology*

Description

morphology

Usage

```
morphology(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species morphology data

Examples

```
 ## Not run:
morphology("Oreochromis niloticus")

## End(Not run)
```

`morphometrics` *morphometrics*

Description

morphometrics

Usage

```
morphometrics(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species morphometrics data

Examples

```
## Not run:
morphometrics("Oreochromis niloticus")

## End(Not run)
```

---

occurrence *occurrence*

Description

occurrence

Usage

```
occurrence()
```

Details

THE OCCURRENCE TABLE HAS BEEN DROPPED BY FISHBASE - THIS FUNCTION NOW RETURNS A STOP MESSAGE.

---

oxygen *oxygen*

Description

oxygen

Usage

```
oxygen(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species oxygen data

Examples

```
 ## Not run:
oxygen("Oreochromis niloticus")

## End(Not run)
```

---

popchar *popchar*

Description

Table of maximum length (Lmax), weight (Wmax) and age (tmax)

Usage

```
popchar(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Details

See references for official documentation. From FishBase.org: This table presents information on maximum length (Lmax), weight (Wmax) and age (tmax) from various localities where a species occurs. The largest values from this table are also entered in the SPECIES table. The POPCHAR table also indicates whether the Lmax, Wmax and tmax values or various combinations thereof refer to the same individual fish.

References

http://www.fishbase.org/manual/english/fishbasethe_popchar_table.htm

Examples

```
## Not run:
popchar("Oreochromis niloticus")

## End(Not run)
```

---

popgrowth *popgrowth*

Description

This table contains information on growth, natural mortality and length at first maturity, which serve as inputs to many fish stock assessment models. The data can also be used to generate empirical relationships between growth parameters or natural mortality estimates, and their correlates (e.g., body shape, temperature, etc.), a line of research that is useful both for stock assessment and for increasing understanding of the evolution of life-history strategies

Usage

```
popgrowth(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of population growth information by species; see details

References

http://www.fishbase.org/manual/english/fishbasethe_popgrowth_table.htm

Examples

```
## Not run:
popgrowth("Oreochromis niloticus")

## End(Not run)
```

---

popqb *popqb*

Description

popqb

Usage

```
popqb(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species popqb

References

http://www.fishbase.org/manual/english/fishbasethe_popqb_table.htm

Examples

```
## Not run:
popqb("Oreochromis niloticus")

## End(Not run)
```

`predators` *predators*

Description

predators

Usage

```
predators(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

`species_list`	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
`fields`	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
`db`	the
`...`	unused; for backwards compatibility only

Value

a table of predators

References

http://www.fishbase.org/manual/english/fishbasethe_predators_table.htm

Examples

```
## Not run:
predators("Oreochromis niloticus")

## End(Not run)
```

`ration` *ration*

Description

ration

Usage

```
ration(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

`species_list`	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
`fields`	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
`db`	the
`...`	unused; for backwards compatibility only

Value

a table of species ration

References

http://www.fishbase.org/manual/english/fishbasethe_ration_table.htm

Examples

```
## Not run:
ration("Oreochromis niloticus")

## End(Not run)
```

`references` *references*

Description

references

Usage

```
references(
  codes = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(),
  ...
)
```

Arguments

`codes`	One or more Fishbase reference numbers, matching the RefNo field
`fields`	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
`db`	the
`...`	unused; for backwards compatibility only

Value

a tibble (data.frame) of reference data

Examples

```
## Not run:
references(codes = 1)
references(codes = 1:6)
references(codes = 1:6, fields = c('Author', 'Year', 'Title'))
references() # all references

## End(Not run)
```

`reproduction` *reproduction*

Description

reproduction

Usage

```
reproduction(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

`species_list`	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
`fields`	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
`server`	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
`version`	a version string for the database, will default to the latest release. see [get_releases()] for details.
`db`	the
`...`	unused; for backwards compatibility only

Value

a table of species reproduction

Examples

```
## Not run:
reproduction("Oreochromis niloticus")

## End(Not run)
```

`sealifebase` *A table of all the the species found in SeaLifeBase, including taxonomic classification and the Species Code (SpecCode) by which the species is identified in SeaLifeBase*

Description

A table of all the the species found in SeaLifeBase, including taxonomic classification and the Species Code (SpecCode) by which the species is identified in SeaLifeBase

Author(s)

Carl Boettiger <carl@ropensci.org>

`spawning` *spawning*

Description

spawning

Usage

```
spawning(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species spawning

Examples

```
## Not run:
spawning("Oreochromis niloticus")

## End(Not run)
```

---

species *species*

Description

Provide wrapper to work with species lists.

Usage

```
species(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Details

The Species table is the heart of FishBase. This function provides a convenient way to query, tidy, and assemble data from that table given an entire list of species. For details, see: http://www.fishbase.org/manual/english/fishb

Species scientific names are defined according to fishbase taxonomy and nomenclature.

Value

a data.frame with rows for species and columns for the fields returned by the query (FishBase 'species' table)

Examples

```
## Not run:

species(c("Labroides bicolor", "Bolbometopon muricatum"))
species(c("Labroides bicolor", "Bolbometopon muricatum"), fields = species_fields$habitat)


## End(Not run)
```

---

species_by_ecosystem *Species list by ecosystem*

Description

Species list by ecosystem

Usage

```
species_by_ecosystem(
  ecosystem,
  species_list = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(),
  ...
)
```

Arguments

ecosystem	(character) an ecosystem name
species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species ecosystems data

Examples

```
## Not run:
species_by_ecosystem(ecosystem = "Arctic", server = "sealifebase")

## End(Not run)
```

---

species_fields *A list of the species_fields available*

Description

A list of the species_fields available

Author(s)

Carl Boettiger <carl@ropensci.org>

species_list *species_list*

Description

Return the a species list given a taxonomic group

Usage

```
species_list(
  Class = NULL,
  Order = NULL,
  Family = NULL,
  Subfamily = NULL,
  Genus = NULL,
  Species = NULL,
  SpecCode = NULL,
  SuperClass = NULL,
  server = getOption("FISHBASE_API", FISHBASE_API)
)
```

Arguments

Class	Request all species in this taxonomic Class
Order	Request all species in this taxonomic Order
Family	Request all species in this taxonomic Family
Subfamily	Request all species in this taxonomic SubFamily
Genus	Request all species in this taxonomic Genus
Species	Request all species in this taxonomic Species
SpecCode	Request species name of species matching this SpecCode
SuperClass	Request all species of this Superclass
server	fishbase or sealifebase

Examples

```
## All species in the Family
  species_list(Family = 'Scaridae')
## All species in the Genus
  species_list(Genus = 'Labroides')
```

species_names *species names*

Description

returns species names given FishBase's SpecCodes

Usage

```
species_names(
  codes,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db()
)
```

Arguments

codes	a vector of speccodes (e.g. column from a table)
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the

Value

A character vector of species names for the SpecCodes

speed *speed*

Description

speed

Usage

```
speed(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species speed data

Examples

```
## Not run:
speed("Oreochromis niloticus")

## End(Not run)
```

---

stocks *stocks*

Description

stocks

Usage

```
stocks(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species stocks data

Examples

```
## Not run:
stocks("Oreochromis niloticus")

## End(Not run)
```

---

swimming *swimming*

Description

swimming

Usage

```
swimming(
  species_list = NULL,
  fields = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(server, version),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. Mostly for backwards compatibility as users can subset by column later
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a table of species swimming data

Examples

```
## Not run:
swimming("Oreochromis niloticus")

## End(Not run)
```

---

synonyms *synonyms*

Description

Check for alternate versions of a scientific name

Usage

```
synonyms(
  species_list = NULL,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Details

For further information on fields returned, see: http://www.fishbase.org/manual/english/fishbasethe_synonyms_table.htm

Value

A table with information about the synonym. Will generally be only a single row if a species name is given. If a FishBase SpecCode is given, all synonyms matching that SpecCode are shown, and the table indicates which one is Valid for FishBase. This may or may not match the valid name for Catalog of Life (Col), also shown in the table. See examples for details.

---

validate_names *validate_names*

Description

Check for alternate versions of a scientific name and return the scientific names FishBase recognizes as valid

Usage

```
validate_names(
  species_list,
  server = getOption("FISHBASE_API", "fishbase"),
  version = get_latest_release(),
  db = default_db(),
  ...
)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, a table for all fish will be returned.
server	can be set to either "fishbase" or "sealifebase" to switch between databases. NOTE: it is usually easier to leave this as NULL and set the source instead using the environmental variable 'FISHBASE_API', e.g. 'Sys.setenv(FISHBASE_API="sealifebase")'.
version	a version string for the database, will default to the latest release. see [get_releases()] for details.
db	the
...	unused; for backwards compatibility only

Value

a string of the validated names

Examples

```
validate_names("Abramites ternetzi")
```

Index