

# Package ‘rfishbase’

August 29, 2016

**Title** R Interface to 'FishBase'

**Description** A programmatic interface to <http://www.fishbase.org>, 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 <http://www.sealifebase.org> data, which contains nearly 200,000 species records for all types of aquatic life not covered by 'FishBase.'

**Version** 2.1.1

**License** CC0

**URL** <https://github.com/ropensci/rfishbase>

**BugReports** <https://github.com/ropensci/rfishbase/issues>

**LazyData** true

**Depends** R (>= 3.0)

**Imports** dplyr (>= 0.4.0), lazyeval, httr (>= 1.0.0), tidyr, methods, utils

**Suggests** devtools, testthat, rmarkdown, knitr

**VignetteBuilder** knitr

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Carl Boettiger [cre, aut],  
Scott Chamberlain [aut],  
Duncan Temple Lang [aut],  
Peter Wainwright [aut]

**Maintainer** Carl Boettiger <cboettig@ropensci.org>

**Repository** CRAN

**Date/Publication** 2016-07-22 01:20:21

**R topics documented:**

rfishbase-package . . . . .	3
common_names . . . . .	3
common_to_sci . . . . .	4
country . . . . .	5
countrysub . . . . .	6
countrysubref . . . . .	7
c_code . . . . .	7
diet . . . . .	8
distribution . . . . .	9
docs . . . . .	10
ecology . . . . .	11
ecosystem . . . . .	12
faoareas . . . . .	13
fecundity . . . . .	14
fishbase . . . . .	15
fooditems . . . . .	15
heartbeat . . . . .	16
introductions . . . . .	17
length_freq . . . . .	18
length_length . . . . .	19
length_weight . . . . .	20
list_fields . . . . .	21
load_taxa . . . . .	22
maturity . . . . .	23
morphology . . . . .	24
morphometrics . . . . .	25
occurrence . . . . .	26
oxygen . . . . .	26
ping . . . . .	27
popchar . . . . .	28
popgrowth . . . . .	29
popqb . . . . .	30
predators . . . . .	31
ration . . . . .	32
reproduction . . . . .	33
sci_to_common . . . . .	34
sealifebase . . . . .	35
spawning . . . . .	35
species . . . . .	36
species_fields . . . . .	37
species_list . . . . .	37
species_names . . . . .	38
speed . . . . .	39
stocks . . . . .	40
swimming . . . . .	41
synonyms . . . . .	42

validate\_names . . . . .

43

Index

44

---

rfishbase-package	<i>The new R interface to Fishbase, v2.0</i>
-------------------	--

---

**Description**

A programmatic interface to <http://www.fishbase.org>, 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 <http://www.sealifebase.org> 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](mailto:carl@ropensci.org)>  
Scott Chamberlain <[scott@ropensci.org](mailto:scott@ropensci.org)>

---

common_names	<i>commonnames</i>
--------------	--------------------

---

**Description**

Return a table of common names

**Usage**

```
common_names(species_list, limit = 1000, server = getOption("FISHBASE_API",
  FISHBASE_API), Language = NULL, fields = c("ComName", "Language",
  "C_Code", "SpecCode"))
```

**Arguments**

- species\_list

A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
- limit

The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
- server

base URL to the FishBase API (by default). For SeaLifeBase, use <http://fishbase.ropensci.org/sealifebase>
- 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. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.

**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.

**Examples**

```
## Not run:
commonnames(c("Labroides bicolor", "Bolbometopon muricatum"))

# subset by English language names
fish <- commonnames("Bolbometopon muricatum")
library(dplyr)
fish %>% filter(Language=="English")

## End(Not run)
```

---

common_to_sci	<i>common_to_sci</i>
---------------	----------------------

---

**Description**

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

**Usage**

```
common_to_sci(x, Language = NULL, limit = 1000,
  server = getOption("FISHBASE_API", FISHBASE_API))
```

**Arguments**

x	a common name or list of common names
Language	a string specifying the language for the common name, e.g. "English"
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>

**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.

Setting the language used explicitly will decrease the data transferred and speed up the function. The limit default is quite high in this call, as it corresponds to the number of common names that match a given species, including different languages and countries.

**Value**

a character vector of scientific names

**See Also**

[commonnames](#), [species\\_list](#), [synonyms](#)

**Examples**

```
## Not run:
common_to_sci(c("Bicolor cleaner wrasse", "humphead parrotfish"), Language="English")
common_to_sci("trout")
common_to_sci(c("trout", "Coho Salmon"))

## End(Not run)
```

---

country	<i>country</i>
---------	----------------

---

**Description**

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

**Usage**

```
country(species_list = NULL, fields = NULL, query = NULL, limit = 200,
        server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

**Details**

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

Examples

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

## End(Not run)
```

---

countrysub	<i>countrysub</i>
------------	-------------------

---

Description

return a table of countrysub for the requested species

Usage

```
countrysub(species_list = NULL, fields = NULL, query = NULL,
  limit = 200, server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

Arguments

- species\_list     A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
- fields           a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.
- query            a named list specifying specific subsets of fields.
- limit            The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
- server           base URL to the FishBase API (by default). For SeaLifeBase, use <http://fishbase.ropensci.org/sealifebase>
- ...              additional arguments to httr::GET

Examples

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

## End(Not run)
```

---

countrysubref	<i>countrysubref</i>
---------------	----------------------

---

**Description**

return a table of countrysubref for the requested species

**Usage**

```
countrysubref(species_list = NULL, fields = NULL, query = NULL,
  limit = 200, server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to httr::GET

**Examples**

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

## End(Not run)
```

---

c_code	<i>c_code</i>
--------	---------------

---

**Description**

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

Usage

```
c_code(c_code, server = getOption("FISHBASE_API", FISHBASE_API),
      fields = "", limit = 500)
```

Arguments

c_code	a C_Code or list of C_Codes (FishBase country code)
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.

Details

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

Examples

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

## End(Not run)
```

---

diet	<i>diet</i>
------	-------------

---

Description

diet

Usage

```
diet(species_list = NULL, fields = NULL, query = NULL, limit = 200,
     server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
--------------	--



fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

**Value**

a table of species diet

**References**

[http://www.fishbase.org/manual/english/fishbasethe\\_diet\\_table.htm](http://www.fishbase.org/manual/english/fishbasethe_diet_table.htm)

**Examples**

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

## End(Not run)
```

---

distribution	<i>distribution</i>
--------------	---------------------

---

**Description**

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

**Usage**

```
distribution(species_list, fields = NULL, server = getOption("FISHBASE_API",
  FISHBASE_API), limit = 500)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.

server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.

### 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	<i>docs</i>
------	-------------

---

### Description

documentation of tables and fields

### Usage

```
docs(table = "", server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

### Arguments

table	the table for which the documentation should be displayed. If no table is given, documentation summarizing all available tables is shown.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

### 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.org and must be manually generated and curated by FishBase.org 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

```
## Not run:
tables <- docs()
# Describe the diet table
dplyr::filter(tables, table == "diet")$description

## End(Not run)
```

---

ecology	<i>ecology</i>
---------	----------------

---

Description

ecology

Usage

```
ecology(species_list = NULL, fields = NULL, query = NULL, limit = 1,
        server = getOption("FISHBASE_API", FISHBASE_API))
```

Arguments

- |              |  |
|--------------|--|
| species_list | A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.                 |
| fields       | a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed. |
| query        | a named list specifying specific subsets of fields.  |
| limit        | The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.  |
| server       | base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>  |

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](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	<i>ecosystem</i>
-----------	------------------

---

## Description

ecosystem

## Usage

```
ecosystem(species_list = NULL, fields = NULL, query = NULL, limit = 200,
          server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

## Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to httr::GET

## Value

a table of species ecosystems data

**Examples**

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

## End(Not run)
```

---

faoareas	<i>faoareas</i>
----------	-----------------

---

**Description**

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

**Usage**

```
faoareas(species_list, fields = NULL, server = getOption("FISHBASE_API",
  FISHBASE_API), limit = 500)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.

**Details**

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

**Examples**

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

## End(Not run)
```

---

fecundity	<i>fecundity</i>
-----------	------------------

---

## Description

fecundity

## Usage

```
fecundity(species_list = NULL, fields = NULL, query = NULL, limit = 200,
          server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

## Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

## Value

a table of species fecundity

## Examples

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

## End(Not run)
```

---

fishbase	<i>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.</i>
----------	---

---

**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>

**References**

[FishBase.org](http://fishbase.org)

---

fooditems	<i>fooditems</i>
-----------	------------------

---

**Description**

fooditems

**Usage**

```
fooditems(species_list = NULL, fields = NULL, query = NULL, limit = 200,
  server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

**Value**

a table of species fooditems

**References**

[http://www.fishbase.org/manual/english/fishbasethe\\_food\\_items\\_table.htm](http://www.fishbase.org/manual/english/fishbasethe_food_items_table.htm)

**Examples**

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

## End(Not run)
```

---

 heartbeat

*heartbeat*


---

**Description**

Check that the FishBase API server is responding

**Usage**

```
heartbeat(server = getOption("FISHBASE_API", FISHBASE_API))
```

**Arguments**

server                      base URL to the FishBase API (by default). For SeaLifeBase, use <http://fishbase.ropensci.org/sealifebase>

**Value**

An httr 'response' object.

**Examples**

```
## Not run:

## Show server response times
resp <- heartbeat()
resp$times

## Show API endpoints:
library("httr")
httr::content(resp)

## End(Not run)
```



---

introductions	<i>introductions</i>
---------------	----------------------

---

## Description

introductions

## Usage

```
introductions(species_list = NULL, fields = NULL, query = NULL,
  limit = 200, server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

## Arguments

<code>species_list</code>	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
<code>fields</code>	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
<code>query</code>	a named list specifying specific subsets of fields.
<code>limit</code>	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
<code>server</code>	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
<code>...</code>	additional arguments to <code>httr::GET</code>

## Value

a table of species introductions data

## Examples

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

## End(Not run)
```

length\_freq

*length\_freq***Description**

return a table of species fooditems

**Usage**

```
length_freq(species_list = NULL, fields = NULL, query = NULL,
            limit = 200, server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

**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	<i>length_length</i>
---------------	----------------------

---

**Description**

return a table of lengths

**Usage**

```
length_length(species_list = NULL, fields = NULL, query = NULL,
              limit = 200, server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

**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-Weight\\_RF\\_JG.pdf](http://www.fishbase.org/manual/english/PDF/FB_Book_CBinohlan_Length-Weight_RF_JG.pdf)

Examples

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

## End(Not run)
```

---

length_weight	<i>length_weight</i>
---------------	----------------------

---

Description

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

Usage

```
length_weight(species_list = NULL, fields = NULL, query = NULL,
  limit = 200, server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

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:

$a'(\text{cm}, \text{g}) = a(\text{mm}, \text{g}) * 10^b$   $a'(\text{cm}, \text{g}) = a(\text{cm}, \text{kg}) * 1000$   $a'(\text{cm}, \text{g}) = a(\text{mm}, \text{mg}) * 10^b / 1000$   $a'(\text{cm}, \text{g}) = a(\text{mm}, \text{kg}) * 10^b * 1000$

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](http://www.fishbase.org/manual/english/fishbasethe_length_weight_table.htm)

### Examples

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

## End(Not run)
```

---

list\_fields

*list\_fields*

---

### Description

list fields

### Usage

```
list_fields(fields, server = getOption("FISHBASE_API", FISHBASE_API),
  implemented_only = TRUE)
```

### Arguments

fields                      field (column name) to search for

server                     base URL to the FishBase API (by default). For SeaLifeBase, use <http://fishbase.ropensci.org/sealifebase>

implemented\_only           by default, only return those tables that have been implemented.

**Details**

method will use partial matching. Hence "Temp" will match column names such as "TempMin" and "TempMax", but "MinTemp" will not. Likewise, neither "Minimum" or "Temperature" will match "TempMin", so begin with the shortest query possible and refine based on search results when necessary. Note also that there is no guarentee that the same column has the same value or same meaning in different tables.

**Value**

a data frame listing the table names (matching function names in rfishbase) and the matching column names those tables have implemented.

**Examples**

```
## Not run:
list_fields("Temp")

## End(Not run)
```

---

load_taxa	<i>load_taxa</i>
-----------	------------------

---

**Description**

Load or update the taxa list

**Usage**

```
load_taxa(update = FALSE, cache = TRUE, server = getOption("FISHBASE_API",
  FISHBASE_API), limit = 5000L)
```

**Arguments**

- |        |   |
|--------|---|
| update | logical, should we query the API to update the available list?  |
| cache  | should we cache the updated version throughout this session? (default TRUE, leave as is)  |
| server | base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a> |
| limit  | The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.         |

**Value**

the taxa list

---

maturity

---

*maturity*


---

## Description

maturity

## Usage

```
maturity(species_list = NULL, fields = NULL, query = NULL, limit = 200,
  server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

## Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

## 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, query = NULL,
            limit = 200, server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

## Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

## 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, query = NULL,
  limit = 200, server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

## Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

## Value

a table of species morphometrics data

## Examples

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

## End(Not run)
```

---

occurrence	<i>occurrence</i>
------------	-------------------

---

**Description**

occurrence

**Usage**

occurrence()

**Details**

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

---

oxygen	<i>oxygen</i>
--------	---------------

---

**Description**

oxygen

**Usage**

```
oxygen(species_list = NULL, fields = NULL, query = NULL, limit = 200,
       server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

- |              |  |
|--------------|--|
| species_list | A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.                 |
| fields       | a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed. |
| query        | a named list specifying specific subsets of fields.  |
| limit        | The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.  |
| server       | base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>  |
| ...          | additional arguments to httr::GET  |

**Value**

a table of species oxygen data

**Examples**

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

## End(Not run)
```

---

ping	<i>ping</i>
------	-------------

---

**Description**

Check that the FishBase MySQL backend to the API is also responding

**Usage**

```
ping(server = getOption("FISHBASE_API", FISHBASE_API))
```

**Arguments**

server                      base URL to the FishBase API (by default). For SeaLifeBase, use <http://fishbase.ropensci.org/sealifebase>

**Value**

An httr 'response' object.

**Examples**

```
## Not run:

## Show server response times
resp <- ping()
resp$times

## End(Not run)
```

---

 popchar

---

*popchar*


---

## Description

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

## Usage

```
popchar(species_list = NULL, fields = NULL, query = NULL, limit = 200,
        server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

## Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

## 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](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, query = NULL, limit = 200,
  server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

**Value**

a table of population growth information by species; see details

**References**

[http://www.fishbase.org/manual/english/fishbasethe\\_popgrowth\\_table.htm](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, query = NULL, limit = 200,
      server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

**Value**

a table of species popqb

**References**

[http://www.fishbase.org/manual/english/fishbasethe\\_popqb\\_table.htm](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, query = NULL, limit = 200,
  server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

**Value**

a table of predators

**References**

[http://www.fishbase.org/manual/english/fishbasethe\\_predators\\_table.htm](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, query = NULL, limit = 200,
       server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

**Value**

a table of species ration

**References**

[http://www.fishbase.org/manual/english/fishbasethe\\_ration\\_table.htm](http://www.fishbase.org/manual/english/fishbasethe_ration_table.htm)

**Examples**

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

## End(Not run)
```



---

reproduction	<i>reproduction</i>
--------------	---------------------

---

## Description

reproduction

## Usage

```
reproduction(species_list = NULL, fields = NULL, query = NULL,
             limit = 200, server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

## Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to <code>httr::GET</code>

## Value

a table of species reproduction

## Examples

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

## End(Not run)
```

---

sci\_to\_common

*sci\_to\_common*


---

## Description

Return the preferred FishBase common name given a scientific name (or speccode)

## Usage

```
sci_to_common(species_list, Language = NULL, limit = 1000,
  server = getOption("FISHBASE_API", FISHBASE_API))
```

## Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
Language	the language for the common name, see details.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>

## Details

If Language is NULL, the common name is the preferred FishBase common name (in English). Otherwise it is the most frequently used common name (which may not be the same as the FishBase common name even with English as the requested Language)

## Value

The common name, if it exists

## Examples

```
## Not run:
sci_to_common("Salmo trutta")
sci_to_common("Salmo trutta", Language="English")
sci_to_common("Salmo trutta", Language="French")

## End(Not run)
```

---

sealifebase	<i>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</i>
-------------	--

---

### 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>

### References

[www.sealifebase.org](http://www.sealifebase.org)

---

spawning	<i>spawning</i>
----------	-----------------

---

### Description

spawning

### Usage

```
spawning(species_list = NULL, fields = NULL, query = NULL, limit = 200,
  server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

### Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
fields	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.
query	a named list specifying specific subsets of fields.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
...	additional arguments to httr::GET

Value

a table of species spawning

Examples

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

## End(Not run)
```

---

species	<i>species</i>
---------	----------------

---

Description

Provide wrapper to work with species lists.

Usage

```
species(species_list = NULL, fields = NULL, query = NULL, limit = 200,
        server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

Arguments

- species\_list    A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
- fields         a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.
- query          a named list specifying specific subsets of fields.
- limit          The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
- server         base URL to the FishBase API (by default). For SeaLifeBase, use <http://fishbase.ropensci.org/sealifebase>
- ...            additional arguments to `httr::GET`

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/fishbase>

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_fields	<i>A list of the species_fields available</i>
----------------	---

---

**Description**

A list of the species\_fields available

**Author(s)**

Carl Boettiger <carl@ropensci.org>

**References**

[FishBase.org](#)

---

species_list	<i>species_list</i>
--------------	---------------------

---

**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,
             SpeciesRefNo = NULL, all_taxa = load_taxa())
```

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
SpeciesRefNo	Request species name of all species matching this SpeciesRefNo
all_taxa	The data.frame of all taxa used for the lookup. By default will be loaded from cache if available, otherwise must be downloaded from the server; about 13 MB, may be slow.

Details

The first time the function is called it will download and cache the complete

Examples

```
## Not run:
## All species in the Family
species_list(Family = 'Scaridae')
## All species in the Genus
species_list(Genus = 'Labroides')

## End(Not run)
```

---

species_names	<i>speciesnames</i>
---------------	---------------------

---

Description

returns species names given FishBase’s SpecCodes

Usage

```
species_names(codes, server = getOption("FISHBASE_API", FISHBASE_API),
  all_taxa = load_taxa(server = server))
```

Arguments

codes	a vector of speccodes (e.g. column from a table)
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
all_taxa	the taxa table, usually loaded by default

**Value**

A character vector of species names for the SpecCodes

---

speed	<i>speed</i>
-------	--------------

---

**Description**

speed

**Usage**

```
speed(species_list = NULL, fields = NULL, query = NULL, limit = 200,
      server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

- |              |  |
|--------------|--|
| species_list | A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.                 |
| fields       | a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed. |
| query        | a named list specifying specific subsets of fields.  |
| limit        | The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.  |
| server       | base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>  |
| ...          | additional arguments to httr::GET  |

**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, query = NULL, limit = 200,
       server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

## Arguments

<code>species_list</code>	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
<code>fields</code>	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.
<code>query</code>	a named list specifying specific subsets of fields.
<code>limit</code>	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
<code>server</code>	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
<code>...</code>	additional arguments to <code>httr::GET</code>

## Value

a table of species stocks data

## Examples

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

## End(Not run)
```



---

swimming	<i>swimming</i>
----------	-----------------

---

**Description**

swimming

**Usage**

```
swimming(species_list = NULL, fields = NULL, query = NULL, limit = 200,
  server = getOption("FISHBASE_API", FISHBASE_API), ...)
```

**Arguments**

- species\_list    A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
- fields         a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transfered over the network if only certain columns are needed.
- query          a named list specifying specific subsets of fields.
- limit          The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
- server         base URL to the FishBase API (by default). For SeaLifeBase, use <http://fishbase.ropensci.org/sealifebase>
- ...            additional arguments to `httr::GET`

**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, limit = 50, server = getOption("FISHBASE_API",
  FISHBASE_API), fields = c("SynGenus", "SynSpecies", "Valid", "Misspelling",
  "Status", "Synonymy", "Combination", "SpecCode", "SynCode", "CoL_ID", "TSN",
  "WoRMS_ID"))
```

## Arguments

<code>species_list</code>	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
<code>limit</code>	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
<code>server</code>	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>
<code>fields</code>	a character vector specifying which fields (columns) should be returned. By default, all available columns recognized by the parser are returned. This option can be used to limit the amount of data transferred over the network if only certain columns are needed.

## Details

For further information on fields returned, see: [http://www.fishbase.org/manual/english/fishbasethe\\_synonyms\\_table.htm](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.

## Examples

```
## Not run:
# Query using a synonym:
synonyms("Callyodon muricatus")

# Check for misspellings or alternate names
x <- synonyms("Labroides dimidatus") # Species name misspelled
```

```

species_list(SpecCode = x$SpecCode) # correct: "Labroides dimidiatus"

# See all synonyms using the SpecCode
species("Bolbometopon muricatum", fields="SpecCode")[[1]]
synonyms(5537)

## End(Not run)

```

---

validate\_names

*validate\_names*


---

## Description

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

## Usage

```

validate_names(species_list, limit = 50, server = getOption("FISHBASE_API",
  FISHBASE_API))

```

## Arguments

species_list	A vector of scientific names (each element as "genus species"). If empty, the table will be loaded up to the limit. This makes it possible to load an entire table without knowledge of the species in the table, thus avoiding unnecessary API calls.
limit	The maximum number of matches from a single API call (e.g. per species). Function will warn if this needs to be increased, otherwise can be left as is.
server	base URL to the FishBase API (by default). For SeaLifeBase, use <a href="http://fishbase.ropensci.org/sealifebase">http://fishbase.ropensci.org/sealifebase</a>

## Value

a string of the validated names

# Index

- \*Topic **data**
  - fishbase, [15](#)
  - sealifebase, [35](#)
  - species\_fields, [37](#)
- \*Topic **package**
  - rfishbase-package, [3](#)
- c\_code, [7](#)
- common\_names, [3](#)
- common\_to\_sci, [4](#)
- commonnames, [5](#)
- commonnames (common\_names), [3](#)
- country, [5](#)
- countrysub, [6](#)
- countrysubref, [7](#)
- diet, [8](#)
- distribution, [9](#)
- docs, [10](#)
- ecology, [11](#)
- ecosystem, [12](#)
- faoareas, [13](#)
- fecundity, [14](#)
- fishbase, [15](#)
- fooditems, [15](#)
- heartbeat, [16](#)
- introductions, [17](#)
- length\_freq, [18](#)
- length\_length, [19](#)
- length\_weight, [20](#)
- list\_fields, [21](#)
- load\_taxa, [22](#)
- maturity, [23](#)
- morphology, [24](#)
- morphometrics, [25](#)
- occurrence, [26](#)
- oxygen, [26](#)
- ping, [27](#)
- popchar, [28](#)
- popgrowth, [29](#)
- poplf (length\_freq), [18](#)
- popll (length\_length), [19](#)
- poplw (length\_weight), [20](#)
- popqb, [30](#)
- predators, [31](#)
- ration, [32](#)
- reproduction, [33](#)
- rfishbase (rfishbase-package), [3](#)
- rfishbase-package, [3](#)
- sci\_to\_common, [34](#)
- sealifebase, [35](#)
- spawning, [35](#)
- species, [36](#)
- species\_fields, [37](#)
- species\_info (species), [36](#)
- species\_list, [5](#), [37](#)
- species\_names, [38](#)
- speciesnames (species\_names), [38](#)
- speed, [39](#)
- stocks, [40](#)
- swimming, [41](#)
- synonyms, [5](#), [42](#)
- validate\_names, [43](#)