

Package ‘icesVocab’

July 22, 2025

Title ICES Vocabularies Database Web Services
Version 1.3.2
Imports icesConnect (>= 1.1.4), memoise, cachem, rlang, httr
Suggests testthat (>= 3.0.0)
Description R interface to access the Vocabularies REST API of the ICES
(International Council for the Exploration of the Sea) Vocabularies database
<<https://vocab.ices.dk/services/>>.
License GPL-3
URL <https://vocab.ices.dk/services/api/swagger/index.html>,
<https://github.com/ices-tools-prod/icesVocab>
RoxygenNote 7.3.2
Encoding UTF-8
BugReports <https://github.com/ices-tools-prod/icesVocab/issues>
Language en-GB
Config/testthat/edition 3
NeedsCompilation no
Author Colin Millar [aut, cre],
Arni Magnusson [aut],
Cedric Briand [ctb]
Maintainer Colin Millar <colin.millar@ices.dk>
Repository CRAN
Date/Publication 2025-05-26 09:10:02 UTC

Contents

icesVocab-package	2
findAphia	3
findCode	4
findCodeType	5

getCodeDetail	6
getCodeList	7
getCodeTypeList	8
getCodeTypeRelation	9
linkCodes	10
vocab_api	11
vocab_get	11
Index	13

icesVocab-package	<i>Vocabularies Database Web Services</i>
-------------------	---

Description

R interface to access the web services of the ICES Vocabularies database of reference codes.

Details

Get full tables:

getCodeTypeList	all code code types
getCodeList	codes of a given type
getCodeDetail	details for a given code

Look up specific codes:

findAphia	species code
---------------------------	--------------

Author(s)

Colin Millar and Arni Magnusson.

References

ICES Vocabularies database: <http://vocab.ices.dk>.
ICES Vocabularies web services: <http://vocab.ices.dk/services/POX.aspx>.

See Also

Useful links:

- <https://vocab.ices.dk/services/api/swagger/index.html>
- <https://github.com/ices-tools-prod/icesVocab>
- Report bugs at <https://github.com/ices-tools-prod/icesVocab/issues>

findAphia*Find Aphia Code*

Description

Look up Aphia code for a given species.

Usage

```
findAphia(species, latin = FALSE, regex = FALSE, full = FALSE)
```

Arguments

species	the species name, either in English (default) or Latin.
latin	whether the species name is in Latin.
regex	whether to match the species name as a regular expression.
full	whether to return a data frame with all species columns.

Details

Species name matches are case-insensitive.

Value

A vector of Aphia codes (default) or a data frame if full is TRUE.

Author(s)

Arni Magnusson.

See Also

[getCodeList](#) can be used to get all Aphia codes; see example on that help page.
[icesVocab-package](#) gives an overview of the package.

Examples

```
## Not run:
findAphia("cod")
findAphia("Gadus morhua", latin = TRUE)

# Multiple matches
findAphia(c("cod", "haddock", "saithe"))

# Regular expressions
findAphia("ray", regex = TRUE, full = TRUE)
findAphia("Sebastes", latin = TRUE, regex = TRUE, full = TRUE)

## End(Not run)
```

`findCode`*Find a Key*

Description

Look up a key for a given code and code type.

Usage

```
findCode(code_type, code, regex = TRUE, full = FALSE)
```

Arguments

<code>code_type</code>	a search string for a code type, e.g. SpecWoRMS, or simply worms.
<code>code</code>	a search string for a code, e.g. a species name, cod, or ship name, Clupea.
<code>regex</code>	whether to match as a regular expression.
<code>full</code>	whether to return a data frame.

Details

Matches are case-insensitive.

Value

A vector of keys (default) or a data frame if full is TRUE.

See Also

[getCodeList](#) can be used to get all code types; see example on that help page.

[icesVocab-package](#) gives an overview of the package.

Examples

```
## Not run:
findCode("aphia", "cod")

# Multiple matches
findCode("aphia", c("cod", "haddock", "saithe"), full = TRUE)

findCodeType("ship", full = TRUE)
findCode("ship", "clupea", full = TRUE)

## End(Not run)
```

findCodeType	<i>Find a Key</i>
--------------	-------------------

Description

Find a lookup key corresponding to a code type.

Usage

```
findCodeType(code, date = NULL, regex = TRUE, full = FALSE)
```

Arguments

code	a code name, e.g. Gear Type, or DATRAS to find all DATRAS related codes.
date	restrict output to code types modified after a given date in yyyy-mm-dd format, e.g. "2010-12-01"
regex	whether to match the stock name as a regular expression.
full	whether to return a data frame with all stock list columns.

Value

A vector of keys (default) or a data frame if full is TRUE.

See Also

[getCodeTypeList](#) gets a list of code types.

[icesVocab-package](#) gives an overview of the package.

Examples

```
## Not run:  
findCodeType("worms")  
  
findCodeType("DATRAS", full = TRUE)  
  
findCodeType("DATRAS", full = TRUE, date = "2010-01-01")  
  
## End(Not run)
```

`getCodeDetail`*Get Details*

Description

Get details for a given code.

Usage

```
getCodeDetail(code_type, code)
```

Arguments

`code_type` the code type, e.g. SpecWoRMS.

`code` the code, e.g. 101170.

Value

A data frame.

See Also

[getCodeTypeList](#) and [getCodeList](#) get code types and codes of a given type.

[icesVocab-package](#) gives an overview of the package.

Examples

```
## Not run:
# Species code 101170 - Myxine glutinosa
getCodeDetail("SpecWoRMS", 101170)

# find details of Haddock using the aphia ID
findCode("species", "haddock", full = TRUE)

getCodeDetail("SpecWoRMS", 126437)

# get info for had-34
getCodeDetail("ICES_StockCode", "had-34")

## End(Not run)
```

`getCodeList`*Get Codes*

Description

Get codes of a given code type.

Usage

```
getCodeList(code_type, code = NULL, date = NULL)
```

Arguments

<code>code_type</code>	the code type, e.g. SpecWoRMS.
<code>code</code>	optional code to restrict output to.
<code>date</code>	restrict output to codes modified after a given date in yyyy-mm-dd format, e.g. "2010-12-01".

Value

A data frame.

See Also

[getCodeTypeList](#) and [getCodeDetail](#) get code types and code details.

[findCodeType](#) and [findAphia](#) provide support for searching the code type and code lists.

[icesVocab-package](#) gives an overview of the package.

Examples

```
## Not run:
# Aphia Species codes
findCodeType("aphia", full = TRUE)
codes <- getCodeList("SpecWoRMS")
head(codes)

findCode("ship", "clupea", full = TRUE)

getCodeList("ICES_Area")
getCodeList("ICES_Area", code = "27.4.a")

## End(Not run)
```

getCodeTypeList	<i>Get Types</i>
-----------------	------------------

Description

Get all code types. Mainly useful as a reminder of what code types are available.

Usage

```
getCodeTypeList(code_type = NULL, date = NULL)
```

Arguments

code_type	optional parameter to restrict output to a specific code type.
date	restrict output to code types modified after a given date in yyyy-mm-dd format, e.g. "2010-12-01"

Value

A data frame.

See Also

[getCodeList](#) and [getCodeDetail](#) get codes of a given type and code details.

[findCodeType](#) searches for a code types based on a search string.

[icesVocab-package](#) gives an overview of the package.

Examples

```
## Not run:
# Find code type for World Register of Marine Species (WoRMS)
types <- getCodeTypeList()
types[grep("worms", tolower(types$Description)),]

# The code type "SpecWoRMS" can be used when calling getCodeTypeList()
getCodeTypeList(code_type = "SpecWoRMS")
# or using the guid
getCodeTypeList(code_type = "f107b677-97ff-4cd9-b4ab-637bb6706066")

findCodeType("worms", full = TRUE)

## End(Not run)
```

getCodeTypeRelation *Get Details*

Description

Get details for a given code.

Usage

```
getCodeTypeRelation(code_type, linked_code_type, rename = TRUE)
```

Arguments

code_type	an ICES Vocabulary code type.
linked_code_type	another ICES Vocabulary code type.
rename	logical, if TRUE (default) the columns of the output is renamed to the code type and linked code type

Value

A data frame.

See Also

[getCodeTypeList](#) and [getCodeList](#) get code types and codes of a given type.
[icesVocab-package](#) gives an overview of the package.

Examples

```
## Not run:
x <- getCodeTypeRelation("Pargroup", "PARAM")

# get all stocks in subarea 4a
# note you would need to cross check with icesSD for
# currently active stocks
lookup <- getCodeTypeRelation("ICES_Area", "ICES_StockCode")
lookup[lookup$ICES_Area == "27.4.a", ]

# get relation between stock code and area, unformatted output
lookup_raw <- getCodeTypeRelation("ICES_StockCode", "ICES_Area", rename = FALSE)
head(lookup_raw)

## End(Not run)
```

linkCodes

*Get Links Between Two Code Types***Description**

Find parent or child linkages between the codes of two code types.

Usage

```
linkCodes(code_type, linked_code_type, codes = NULL, ...)
```

Arguments

code_type an ICES Vocabulary code type.
 linked_code_type another ICES Vocabulary code type.
 codes a vector of codes to restrict the output to.
 ... for back compatibility.

Value

A data frame.

Author(s)

Colin Millar.

See Also

[icesVocab-package](#) gives an overview of the package.

Examples

```
## Not run:
x <- linkCodes("Pargroup", "PARAM")
# note it is much faster to use:
# getCodeTypeRelation("Pargroup", "PARAM")
# when getting a full lookup table.

# get all areas for cod.27.1-2
linkCodes("ICES_StockCode", "ICES_Area", codes = c("cod.27.1-2"))

# get areas for cod.27.21, cod.27.24-32 and cod.27.46a7d20
linkCodes("ICES_StockCode", "ICES_Area", codes = c("cod.27.21", "cod.27.24-32", "cod.27.46a7d20"))

## End(Not run)
```

vocab_api	<i>Build a SAG web service url</i>
-----------	------------------------------------

Description

utility to build a url with optional query arguments

Usage

```
vocab_api(service, ...)
```

Arguments

service	the name of the service
...	name arguments will be added as queries

Value

a complete url as a character string

Examples

```
vocab_api("hi", bye = 21)
vocab_api("CodeType")
```

vocab_get	<i>Get a url</i>
-----------	------------------

Description

Get a url, optionally using an ICES authentication token

Usage

```
vocab_get(
  url,
  retry = TRUE,
  quiet = !getOption("icesVocab.messages"),
  verbose = FALSE,
  content = TRUE
)

vocab_get_cached(
  url,
  retry = TRUE,
```

```
quiet = !getOption("icesVocab.messages"),  
verbose = FALSE,  
content = TRUE  
)
```

Arguments

url	the url to get.
retry	should the get request be retried if first attempt fails? default TRUE.
quiet	should all messages be suppressed, default FALSE.
verbose	should verbose output from the http request be returned? default FALSE.
content	should content be returned, or the full http response? default TRUE, i.e. content is returned by default.

Value

content or an http response.

Functions

- `vocab_get_cached()`: cached version of `vocab_get`

See Also

[vocab_api](#) builds a SAG web service url.
[icesVocab-package](#) gives an overview of the package.

Examples

```
## Not run:  
vocab_get(vocab_api("StockList", year = 2022))  
  
## End(Not run)
```

Index

`findAphia`, [2](#), [3](#), [7](#)

`findCode`, [4](#)

`findCodeType`, [5](#), [7](#), [8](#)

`getCodeDetail`, [2](#), [6](#), [7](#), [8](#)

`getCodeList`, [2–4](#), [6](#), [7](#), [8](#), [9](#)

`getCodeTypeList`, [2](#), [5–7](#), [8](#), [9](#)

`getCodeTypeRelation`, [9](#)

`icesVocab (icesVocab-package)`, [2](#)

`icesVocab-package`, [2](#)

`linkCodes`, [10](#)

`vocab_api`, [11](#), [12](#)

`vocab_get`, [11](#)

`vocab_get_cached (vocab_get)`, [11](#)