Package: icesSAG (via r-universe)

August 17, 2024

Version 1.4.1

Title Stock Assessment Graphs Database Web Services

Depends R (>= 2.10)

Imports png, httr, xml2, openssl, icesVocab, icesConnect, memoise, cachem, rlang

Suggests testthat

Description R interface to access the web services of the ICES Stock Assessment Graphs database <https://sg.ices.dk>.

License GPL (>= 2)

URL https://sg.ices.dk, https://github.com/ices-tools-prod/icesSAG

BugReports https://github.com/ices-tools-prod/icesSAG/issues

RoxygenNote 7.2.3

Encoding UTF-8

Repository https://ices-tools-prod.r-universe.dev

RemoteUrl https://github.com/ices-tools-prod/icesSAG

RemoteRef HEAD

RemoteSha 79da39d8c65baca566d17ec1fc8bfb3e2c8fa16e

Contents

`icesSAG-package` *Stock Assessment Graphs Database Web Services*

Description

R interface to access the web services of the ICES Stock Assessment Graphs database.

Details

Get dataset:

`getSummaryTable`	summary results
`getFishStockReferencePoints`	reference points
`getSAG`	any data

Look up codes:

`findAssessmentKey`	find assessment key
`getListStocks`	list of stocks

Author(s)

Colin Millar, Scott Large, and Arni Magnusson.

References

ICES Stock Assessment Graphs database: http://sg.ices.dk.

ICES Stock Assessment Graphs web services: http://sg.ices.dk/webservices.aspx.

---

convertSAGxml *Create and read the SAG XML data transfer file*

Description

Convert between R data (a list and a data.frame) and the XML format required for uploading data to the SAG database.

Usage

```
createSAGxml(info, fishdata)

readSAGxml(file)
```

Arguments

info	a list of stock information
fishdata	a data frame of fish data
file	an xml file name

Value

Either a list containing info and fishdata, or a string containing the xml file.

See Also

stockInfo creates a list of stock information.

stockFishdata creates a data frame of fish stock summary data.

Examples

```
info <- stockInfo(StockCode = "cod.27.347d",
                  AssessmentYear = 2017,
                  StockCategory = 1,
                  ModelType = "A",
                  ModelName = "SCA",
                  ContactPerson = "itsme@fisheries.com")
fishdata <- stockFishdata(Year = 1990:2017, Catches = 100)
xmlfile <- createSAGxml(info, fishdata)

out <- readSAGxml(xmlfile)
```

`findAssessmentKey` *Find a Key*

Description

Find a lookup key corresponding to a stock in a given assessment year.

Usage

```
findAssessmentKey(
  stock = NULL,
  year = 0,
  published = TRUE,
  regex = TRUE,
  full = FALSE
)
```

Arguments

`stock`	a stock name, e.g. cod-347d, or cod to find all cod stocks, or NULL (default) to process all stocks.
`year`	the assessment year, e.g. 2015, or 0 to process all years.
`published`	whether to include only years where status is "Published" (applies only when non-secure web services are in use, secure web service always returns unpublished stocks).
`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.

Author(s)

Arni Magnusson and Colin Millar.

See Also

`StockList` gets a list of stocks.

`icesSAG-package` gives an overview of the package.

Examples

```
## Not run:
findAssessmentKey("had.27.46a20", 2023, full = TRUE)

## End(Not run)
```

`FishStockReferencePoints` *Get Reference Points*

Description

Get biological reference points for all stocks in a given assessment year.

Usage

```
FishStockReferencePoints(assessmentKey, ...)
```

Arguments

`assessmentKey`	the unique identifier of the stock assessment
`...`	arguments passed to `ices_get`.

Value

A data frame.

Author(s)

Colin Millar.

See Also

`getSAG` supports querying many years and quarters in one function call.

`StockList` and `SummaryTable` get a list of stocks and summary results.

`icesSAG-package` gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("cod.27.21", year = 2023)
refpts <- FishStockReferencePoints(assessmentKey)
refpts

# To get all reference points in a given assessment year:
keys2022 <- findAssessmentKey(year = 2022, full = TRUE)
keys2022 <- keys2022[keys2022$Purpose == "Advice",]
refpts2022 <- FishStockReferencePoints(keys2022$AssessmentKey)
refpts2022

## End(Not run)
```

`getCustomColumns` *Get the Custom Columns for SAG records*

Description

Get custom columns, such as alternative biomass series or Fproxy reference points for records in the SAG database.

Usage

```
getCustomColumns(assessmentKey, ...)
```

Arguments

`assessmentKey`	the unique identifier of the stock assessment
`...`	arguments passed to `ices_get`.

Value

A data frame.

Author(s)

Colin Millar.

See Also

`getSAG` supports querying many years and quarters in one function call.

`StockList` and `FishStockReferencePoints` get a list of stocks and reference points.

`icesSAG-package` gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("bli.27.5a14")
customs <- getCustomColumns(assessmentKey)
head(customs)

## End(Not run)
```

getFishStockReferencePoints *Get Reference Points*

Description

Get biological reference points for all stocks in a given assessment year.

Usage

```
getFishStockReferencePoints(assessmentKey, ...)
```

Arguments

assessmentKey	the unique identifier of the stock assessment
...	to allow scope for back compatibility

Value

A data frame.

Author(s)

Colin Millar and Scott Large.

See Also

getSAG supports querying many years and quarters in one function call.

getListStocks and getSummaryTable get a list of stocks and summary results.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("cod-2224", year = 2016)
refpts <- getFishStockReferencePoints(assessmentKey)
refpts

#To get all reference points in a given assessment year:
keys2016 <- findAssessmentKey(year = 2016)
refpts2016 <- getFishStockReferencePoints(keys2016)
refpts2016

## End(Not run)
```

getLatestStockAdviceList *Get List of Most Recent Advice*

Description

Get a list of the most recent advice for all fish stocks.

Usage

```
getLatestStockAdviceList()
```

Value

A data frame.

Author(s)

Colin Millar, Scott Large, and Arni Magnusson.

See Also

`getSummaryTable` gets a summary table of historical stock size.

`getFishStockReferencePoints` gets biological reference points.

`icesSAG-package` gives an overview of the package.

Examples

```
## Not run:
stocks <- getLatestStockAdviceList()

## End(Not run)
```

getListStocks *Get a List of Fish Stocks*

Description

Get a list of fish stocks for a given assessment year.

Usage

```
getListStocks(year)
```

Arguments

year	the assessment year, e.g. 2015, or 0 to process all years.

Value

A data frame.

Author(s)

Colin Millar, Scott Large, and Arni Magnusson.

See Also

getSummaryTable gets a summary table of historical stock size.

getFishStockReferencePoints gets biological reference points.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
stocks <- getListStocks(2015)

## End(Not run)
```

---

getSAG *Get Any SAG Data*

Description

This function combines the functionality of getListStocks, getFishStockReferencePoints, and getSummaryTable. It supports querying many stocks and years in one function call.

Usage

```
getSAG(stock, year, data = "summary", combine = TRUE, purpose = "Advice")
```

Arguments

stock	a stock name, e.g. cod-347d, or cod to find all cod stocks, or NULL to process all stocks.
year	the assessment year, e.g. 2015, or 0 to process all years.
data	the data of interest, either "summary", "refpts" or "source".
combine	whether to combine the list output to a data frame.
purpose	the purpose of the entry, options are "Advice", "Bench", "InitAdvice", default is "Advice".

Value

A data frame (default) or a list if `combine` is `TRUE`.

Note

Only years with "Published" status are returned.

Author(s)

Colin Millar.

See Also

`StockList`, `SummaryTable`, and `FishStockReferencePoints` get a list of stocks, summary results, and reference points.

`findAssessmentKey` finds lookup keys.

`icesSAG-package` gives an overview of the package.

Examples

```
## Not run:
summary <- getSAG("had.27.46a20", 2022)
refpts <- getSAG("had.27.46a20", 2022, "refpts")

cod_summary <- getSAG("cod", 2022)
cod_refpts <- getSAG("cod", 2015:2016, "refpts")
cod_data <- getSAG("cod", 2017, "source-data")

## End(Not run)
```

`getSAGGraphs` *Get Summary Graphs of Stock Assessment Output*

Description

Get summary graphs of catches, recruitment, fishing pressure, and spawning stock biomass.

Usage

```
getSAGGraphs(assessmentKey, ...)
```

Arguments

`assessmentKey`	the unique identifier of the stock assessment
`...`	to allow scope for back compatibility

Value

An array representing a bitmap.

Author(s)

Colin Millar and Scott Large.

See Also

`getListStocks` gets a list of stocks.

`getFishStockReferencePoints` gets biological reference points.

`icesSAG-package` gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("cod", 2015)
graphs <- getSAGGraphs(assessmentKey[1])
plot(graphs)
# note this stock only has one graph see:
# http://standardgraphs.ices.dk/ViewCharts.aspx?key=8309

## End(Not run)
```

---

getSAGGTypegraphsandSettings

Get Details on SAG Charts and Settings

Description

List all possible chart settings for each chart type (0 = general, 1 = Landings etc.).

Usage

```
getSAGTypeGraphs()

getSAGTypeSettings(SAGChartKey)
```

Arguments

`SAGChartKey`	the type identifier of the SAG chart, e.g. 0, 1, 2, ...

Value

a data frame with SAG chart type IDs and settings IDs.

Examples

```
## Not run:
getSAGTypeGraphs()

getSAGTypeSettings(0)[-4]

## End(Not run)
```

---

getsetSAGSettingsForAStock

Get and Set SAG Chart Settings

Description

details

Usage

```
getSAGSettingsForAStock(assessmentKey)

setSAGSettingForAStock(
  assessmentKey,
  chartKey,
  settingKey,
  settingValue,
  copyNextYear
)
```

Arguments

assessmentKey	the unique identifier of the stock assessment
chartKey	the type identifier of the SAG chart, e.g. 0, 1, 2, ...
settingKey	the type identifier of the SAG chart setting, e.g. 0, 1, 2, ...
settingValue	the vale of the setting
copyNextYear	should the settings be copied to next year (TRUE) or not (FALSE)

Value

A data frame with SAG chart type IDs, settings IDs and setting values.

Examples

```
## Not run:
key <- findAssessmentKey("cod.21.1", 2017)
graphs <- getSAGGraphs(key[1])
plot(graphs)
getSAGSettingsForAStock(key [1])
chart1 <- getLandingsGraph(key [1])
setSAGSettingForAStock(key [2], 1, 1, "Catches of cod.21.1 in 2017",
FALSE)
setSAGSettingForAStock(key [2], 1, 11, 10,
FALSE)
plot(chart1)
chart2 <- getSpawningStockBiomassGraph(key [1])
plot(chart2)
setSAGSettingForAStock(key [1], 4, 1, "SSB of cod.21.1 in 2017",
FALSE)
plot(chart2)

## End(Not run)
```

`getsetStockChartSettings` *Get and Set SAG Chart Settings*

Description

details

Usage

```
StockSettings(assessmentKey, ...)

setStockSettings(
  assessmentKey,
  chartKey,
  settingKey,
  settingValue,
  copyNextYear,
  ...
)
```

Arguments

`assessmentKey`	the unique identifier of the stock assessment
`...`	arguments passed to `ices_get`.
`chartKey`	the type identifier of the SAG chart, e.g. 0, 1, 2, ...

settingKey	the type identifier of the SAG chart setting, e.g. 0, 1, 2, ...
settingValue	the vale of the setting
copyNextYear	should the settings be copied to next year (TRUE) or not (FALSE)

Value

A data frame with SAG chart type IDs, settings IDs and setting values.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("had.27.46a20", year = 2022)

StockSettings(assessmentKey)

## End(Not run)
```

---

getStandardAssessmentGraphs

Get a Graph of Stock Assessment Output

Description

Get a graph of stock assessment output, e.g., historical stock size, recruitment, and fishing pressure.

Usage

```
getLandingsGraph(assessmentKey, ...)

getRecruitmentGraph(assessmentKey, ...)

getFishingMortalityGraph(assessmentKey, ...)

getSpawningStockBiomassGraph(assessmentKey, ...)

getFishMortality(assessmentKey, ...)

getstock_recruitment(assessmentKey, ...)

getYSSB(assessmentKey, ...)

getSSBHistoricalPerformance(assessmentKey, ...)

getFishingMortalityHistoricalPerformance(assessmentKey, ...)

getRecruitmentHistoricalPerformance(assessmentKey, ...)

getStockStatusTable(assessmentKey, ...)
```

Arguments

assessmentKey	the unique identifier of the stock assessment
...	to allow scope for back compatibility

Value

An array representing a bitmap.

See Also

getListStocks gets a list of stocks.

getFishStockReferencePoints gets biological reference points.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
assessmentKeys <- findAssessmentKey("had", 2015)
landings_img <- getLandingsGraph(assessmentKeys[1])
plot(landings_img)

landings_plots <- getLandingsGraph(assessmentKeys)
plot(landings_plots)

## End(Not run)
```

---

getStockSourceData *Get Source Data*

Description

Get a copy of the source data for the specified stocks.

Usage

```
getStockDownloadData(assessmentKey, ...)

getStockSourceData(assessmentKey, ...)
```

Arguments

assessmentKey	the unique identifier of the stock assessment
...	to allow scope for back compatibility

Value

A data frame.

Author(s)

Colin Millar.

See Also

getSAG supports querying many years in one function call.

getListStocks and getFishStockReferencePoints get a list of stocks and reference points.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("cod-2224", year = 2016)
sourcedat <- getStockDownloadData(assessmentKey)
head(sourcedat[[1]])

## End(Not run)
```

---

getStockStatusValues *Get the Values in a Stock Status Table*

Description

Get summary results of historical stock size, recruitment, and fishing pressure.

Usage

```
getStockStatusValues(assessmentKey, ...)
```

Arguments

assessmentKey	the unique identifier of the stock assessment
...	to allow scope for back compatibility

Value

A data frame.

Author(s)

Colin Millar.

See Also

getSAG supports querying many years and quarters in one function call.

getListStocks and getFishStockReferencePoints get a list of stocks and reference points.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("cod-2224", year = 2016)
status <- getStockStatusValues(assessmentKey)
status

## End(Not run)
```

`getSummaryTable` *Get a Summary Table of Historical Stock Size*

Description

Get summary results of historical stock size, recruitment, and fishing pressure.

Usage

```
getSummaryTable(assessmentKey, ...)
```

Arguments

`assessmentKey`	the unique identifier of the stock assessment
`...`	to allow scope for back compatibility

Value

A data frame.

Author(s)

Colin Millar and Scott Large.

See Also

`getSAG` supports querying many years and quarters in one function call.

`getListStocks` and `getFishStockReferencePoints` get a list of stocks and reference points.

`icesSAG-package` gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("cod-2224", year = 2016)
sumtab <- getSummaryTable(assessmentKey)
head(sumtab)
attributes(sumtab)$notes

## End(Not run)
```

getTokenExpiration *Get the expiration time of a Standard Graphs token.*

Description

Get the number of days remaining for a standard graphs personal access token.

Usage

```
getTokenExpiration()
```

Value

An integer.

Author(s)

Colin Millar.

Examples

```
## Not run:
getTokenExpiration()

## End(Not run)

# tell icesSAG to use the SG token
options(icesSAG.use_token = TRUE)
```

getYSBRSummaryTable *Get a Summary Table of Yield and Spawning Biomass Per Recruit*

Description

Get summary results of historical stock size, recruitment, and fishing pressure.

Usage

```
getYSBRSummaryTable(assessmentKey, ...)
```

Arguments

assessmentKey	the unique identifier of the stock assessment
...	to allow scope for back compatibility

Value

A data frame.

Author(s)

Colin Millar and Scott Large.

See Also

`getSAG` supports querying many years and quarters in one function call.

`getListStocks` and `getFishStockReferencePoints` get a list of stocks and reference points.

`icesSAG-package` gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("cod-2224", year = 2015)
sumtab <- getYSBRSummaryTable(assessmentKey)
head(sumtab)

## End(Not run)
```

---

get_custom_plot *Get a Custom Graph of Stock Assessment Output*

Description

Get a custom graph of stock assessment output.

Usage

```
get_custom_plot(assessmentKey, type = c(15, 16, 17, 18), width = 800, ...)
```

Arguments

`assessmentKey`	the unique identifier of the stock assessment
`type`	The type of plot: values can be 15, 16, 17, or 18.
`width`	width of the image in pixels
`...`	to allow scope for back compatibility

Value

An array representing a bitmap.

See Also

StockList gets a list of stocks.

FishStockReferencePoints gets biological reference points.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("ane.27.9a", year = 2022)
landings_img <- get_custom_plot(assessmentKey, "15")

plot(landings_img)

landings_plots <- get_image(assessmentKey, "landings")
plot(landings_plots)

## End(Not run)
```

---

get_image *Get a Graph of Stock Assessment Output*

Description

Get a graph of stock assessment output, e.g., historical stock size, recruitment, and fishing pressure.

Usage

```
get_image(
  assessmentKey,
  type = c("landings", "recruitment", "ssb", "mortality", "historical_mortality",
    "historical_ssb", "historical_recruitment"),
  width = 800,
  ...
)
```

Arguments

assessmentKey	the unique identifier of the stock assessment
type	The type of plot: values can be landings, recruitment ssb, mortality, historical_mortality, historical_ssb, historical_recruitment.
width	width of the image in pixels
...	to allow scope for back compatibility

Value

An array representing a bitmap.

See Also

StockList gets a list of stocks.

FishStockReferencePoints gets biological reference points.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("cod.27.21", year = 2022:2023)
landings_img <- get_image(assessmentKey[1], "landings")

plot(landings_img)

landings_plots <- get_image(assessmentKey, "landings")
plot(landings_plots)

## End(Not run)
```

---

ices_get *Get a url*

Description

Get a url, optionally using an ICES authentication token

Usage

```
ices_get(
  url,
  retry = TRUE,
  quiet = FALSE,
  verbose = FALSE,
  content = TRUE,
  use_token = getOption("icesSAG.use_token")
)

ices_get_cached(
  url,
  retry = TRUE,
  quiet = FALSE,
  verbose = FALSE,
  content = TRUE,
  use_token = getOption("icesSAG.use_token")
)
```

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 form the http request be returned? default FALSE.
content	should content be returned, or the full http reponse? default TRUE, i.e. content is returned by default.
use_token	should an authentication token be sent with the request? default is the value of the option icesSAG.use_token.

Value

content or an http response.

Functions

- ices_get_cached(): cached version of ices_get

See Also

sag_api builds a SAG web service url.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
ices_get(sag_api("StockList", year = 2022))

## End(Not run)
```

LatestStockAdviceList *Get a List of the Latest Fish Stock Assessments*

Description

Get a list of the latest fish stock assessments. Only published stocks are returned.

Usage

```
LatestStockAdviceList(...)
```

Arguments

...	arguments passed to ices_get.

Value

A data frame.

See Also

StockList Get a list of fish stock assessments.

getFishStockReferencePoints gets biological reference points.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
stock_list <- LatestStockAdviceList()

## End(Not run)
```

SAGTypeChartsandSettings

Get Details on SAG Charts and Settings

Description

List all possible chart settings for each chart type (0 = general, 1 = Landings etc.).

Usage

```
SAGTypeCharts(...)

SettingsForChartType(SAGChartKey, ...)
```

Arguments

...	arguments passed to ices_get.
SAGChartKey	the type identifier of the SAG chart, e.g. 0, 1, 2, ...

Value

a data frame with SAG chart type IDs and settings IDs.

Examples

```
## Not run:
SAGTypeCharts()

SettingsForChartType(0)[-4]

## End(Not run)
```

sag_api *Build a SAG web service url*

Description

utility to build a url with optional query arguments

Usage

```
sag_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

```
sag_api("hi", bye = 21)
sag_api("StockList", year = 2021)
```

StockDownload *Get Source Data*

Description

Get a copy of the source data for the specified stocks.

Usage

```
StockDownload(assessmentKey, ...)
```

Arguments

assessmentKey	the unique identifier of the stock assessment, can be a vector
...	arguments passed to ices_get.

Value

A data frame.

Author(s)

Colin Millar.

See Also

`getSAG` supports querying many years in one function call.

`StockList` and `getFishStockReferencePoints` get a list of stocks and reference points.

`icesSAG-package` gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("had.27.46a20", year = 2022)
sourcedat <- StockDownload(assessmentKey)
head(sourcedat)

## End(Not run)
```

---

stockFishdata *Create a data.frame of fish stock data*

Description

This function is a wrapper to `data.frame(...)` in which the names are forced to match with the names required for the SAG database. See http://dome.ices.dk/datsu/selRep.aspx?Dataset=126 for more details.

Usage

```
stockFishdata(Year, ...)
```

Arguments

`Year`	a vector of years.
`...`	additional information, e.g. Recruitment, StockSize, Landings, ...

Value

A data.frame, where all names are valid column names in the SAG database.

Author(s)

Colin Millar.

Examples

```
stockFishdata(Year = 1990:2017, Catches = 100)
```

stockInfo *Create a list of fish stock information*

Description

This function is a wrapper to `list(...)` in which the names are forced to match with the names required for the SAG database. See http://dome.ices.dk/datsu/selRep.aspx?Dataset=126 for more details.

Usage

```
stockInfo(
  StockCode,
  AssessmentYear,
  ContactPerson,
  StockCategory,
  Purpose = "Advice",
  ModelType,
  ModelName,
  ...
)
```

Arguments

StockCode	a stock name, e.g. cod-347d.
AssessmentYear	the assessment year, e.g. 2015.
ContactPerson	the email for the person responsible for uploading the stock data.
StockCategory	Category of the assessment used (see below)
Purpose	the purpose of the entry, options are "Advice", "Bench", "InitAdvice", default is "Advice".
ModelType	the type of the model used (see below for links to more information)
ModelName	the name (acronym) of the model used if available (see below for links to more information)
...	additional information, e.g. BMGT, FMSY, RecruitmentAge, ...

Value

A named sag.list, inheriting from a list, where all names are valid column names in the SAG database.

Author(s)

Colin Millar.

See Also

Links to the relevant ICES vocabularies list are here StockCode: https://vocab.ices.dk/?ref=357 StockCategory: https://vocab.ices.dk/?ref=1526 Purpose: https://vocab.ices.dk/?ref=1516 ModelType: https://vocab.ices.dk/?ref=1524 ModelName: https://vocab.ices.dk/?ref=1525

Link to the relevant format description is https://datsu.ices.dk/web/selRep.aspx?Dataset=126

Examples

```
info <-
  stockInfo(StockCode = "cod.27.47d20",
            AssessmentYear = 2017,
            StockCategory = 1,
            ModelType = "A",
            ModelName = "SCA",
            ContactPerson = "itsme@fisheries.com")

 info
 info$mistake <- "oops"
 info
 # should have gotten a warning message

 ## Not run:
 # use icesVocab to list valid codes etc.
 library(icesVocab)
 # print the list of valid stock codes
 stock.codes <- getCodeList("ICES_StockCode")
 stock.codes[1:10,1:2]

 # print the list of assessment model types in the ICES vocabulary
 model.types <- getCodeList("AssessmentModelType")
 model.types[1:2]

 # print the list of assessment model names in the ICES vocabulary
 model.names <- getCodeList("AssessmentModelName")
 model.names$Key

## End(Not run)
```

StockList *Get a List of Fish Stock Assessments*

Description

Get a list of fish stock assessments for a given assessment year. If an authentication token is not provided, only published stocks are returned.

Usage

```
StockList(year, modifiedAfter = NULL, ...)
```

Arguments

year	the assessment year, e.g. 2022, or 0 to process all years.
modifiedAfter	date-time parameter in the format "". If set will only return stocks assessments modified after the provided date.
...	arguments passed to ices_get.

Value

A data frame.

See Also

getSummaryTable gets a summary table of historical stock size.

getFishStockReferencePoints gets biological reference points.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
 stock_list <- StockList(year = 2022)

## End(Not run)
```

StockStatusValues *Get the Values in a Stock Status Table*

Description

Get summary results of historical stock size, recruitment, and fishing pressure.

Usage

```
StockStatusValues(assessmentKey, ...)
```

Arguments

assessmentKey	the unique identifier of the stock assessment
...	arguments passed to ices_get.

Value

A data frame.

Author(s)

Colin Millar.

See Also

getSAG supports querying many years and quarters in one function call.

StockList and FishStockReferencePoints get a list of stocks and reference points.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("had.27.46a20", year = 2022)
status <- StockStatusValues(assessmentKey)
status

## End(Not run)
```

SummaryTable *Get a Summary Table of Historical Stock Size*

Description

Get summary results of historical stock size, recruitment, and fishing pressure.

Usage

```
SummaryTable(assessmentKey, ...)
```

Arguments

assessmentKey	the unique identifier of the stock assessment
...	arguments passed to ices_get.

Value

A data frame.

Author(s)

Colin Millar.

See Also

getSAG supports querying many years and quarters in one function call.

StockList and FishStockReferencePoints get a list of stocks and reference points.

icesSAG-package gives an overview of the package.

Examples

```
## Not run:
assessmentKey <- findAssessmentKey("had.27.46a20", year = 2022)
sumtab <- SummaryTable(assessmentKey)
head(sumtab)

## End(Not run)
```

`uploadStock` *Upload New or Updated Fish Stock Assessment Results*

Description

Get summary results of historical stock size, recruitment, and fishing pressure.

Usage

```
uploadStock(info, fishdata, verbose = FALSE)
```

Arguments

`info`	a list of stock information
`fishdata`	a data.frame of fish data
`verbose`	if TRUE more verbose messages are reported

Value

The database key of the new / updated stock, or 0 if there was an error.

Author(s)

Colin Millar.

See Also

`stockInfo` creates a list of stock information.

`stockFishdata` creates a data.frame of fish stock summary data.

Index