

# Package ‘icesSAG’

February 17, 2022

**Version** 1.4.0

**Title** Stock Assessment Graphs Database Web Services

**Depends** R (>= 2.10)

**Imports** png, httr, xml2, openssl, icesVocab

**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** 7.1.2

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Colin Millar [aut, cre],  
Scott Large [aut],  
Arni Magnusson [aut],  
Carlos Pinto [aut]

**Maintainer** Colin Millar <[colin.millar@ices.dk](mailto:colin.millar@ices.dk)>

**Repository** CRAN

**Date/Publication** 2022-02-17 18:12:14 UTC

## R topics documented:

icesSAG-package . . . . .	2
convertSAGxml . . . . .	3
findAssessmentKey . . . . .	4
getCustomColumns . . . . .	5
getFishStockReferencePoints . . . . .	6
getLatestStockAdviceList . . . . .	7
getListStocks . . . . .	7

getSAG . . . . .	8
getSAGGraphs . . . . .	9
getSAGTypegraphsandSettings . . . . .	10
getsetSAGSettingsForAStock . . . . .	11
getStandardAssessmentGraphs . . . . .	12
getStockSourceData . . . . .	13
getStockStatusValues . . . . .	14
getSummaryTable . . . . .	15
getTokenExpiration . . . . .	16
getYSBRSummaryTable . . . . .	17
stockFishdata . . . . .	18
stockInfo . . . . .	18
uploadStock . . . . .	20

<b>Index</b>	<b>22</b>
--------------	-----------

---

icesSAG-package	<i>Stock Assessment Graphs Database Web Services</i>
-----------------	--

---

## Description

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

## Details

*Get dataset:*

<code>getSummaryTable</code>	summary results
<code>getFishStockReferencePoints</code>	reference points
<code>getSAG</code>	any data

*Look up codes:*

<code>findAssessmentKey</code>	find assessment key
<code>getListStocks</code>	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,
                 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  
)
```

```
findKey(stock, 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

[getListStocks](#) gets a list of stocks.

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

### Examples

```
## Not run:  
findAssessmentKey("cod-347d", 2015, full = TRUE)  
  
## 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 a record in the SAG database.

### Usage

```
getCustomColumns(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("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)**

Arni Magnusson and Colin Millar.

**See Also**

[getListStocks](#), [getSummaryTable](#), and [getFishStockReferencePoints](#) 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("cod-347d", 2015)
refpts <- getSAG("cod-347d", 2015, "refpts")

getSAG("her.27.3a47d", 2017, "refpts", purpose = "Benchmark")

cod_summary <- getSAG("cod", 2015)
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 value 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)
```

---

```
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)
```

---

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: <http://vocab.ices.dk/?ref=357> StockCategory: <http://vocab.ices.dk/?ref=1526> Purpose: <http://vocab.ices.dk/?ref=1516> ModelType: <http://vocab.ices.dk/?ref=1524> ModelName: <http://vocab.ices.dk/?ref=1525>

Link to the relevant format description is <http://datsu.ices.dk/web/relRep.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)

```

---

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

convertSAGxml, [3](#)  
createSAGxml (convertSAGxml), [3](#)  
  
findAssessmentKey, [2](#), [4](#), [9](#)  
findKey (findAssessmentKey), [4](#)  
  
getCustomColumns, [5](#)  
getFishingMortalityGraph  
    (getStandardAssessmentGraphs),  
    [12](#)  
getFishingMortalityHistoricalPerformance  
    (getStandardAssessmentGraphs),  
    [12](#)  
getFishMortality  
    (getStandardAssessmentGraphs),  
    [12](#)  
getFishStockReferencePoints, [2](#), [5](#), [6](#),  
    [7–10](#), [13–15](#), [17](#)  
getLandingsGraph  
    (getStandardAssessmentGraphs),  
    [12](#)  
getLatestStockAdviceList, [7](#)  
getListStocks, [2](#), [4–6](#), [7](#), [9](#), [10](#), [13–15](#), [17](#)  
getRecruitmentGraph  
    (getStandardAssessmentGraphs),  
    [12](#)  
getRecruitmentHistoricalPerformance  
    (getStandardAssessmentGraphs),  
    [12](#)  
getSAG, [2](#), [5](#), [6](#), [8](#), [14](#), [15](#), [17](#)  
getSAGGraphs, [9](#)  
getSAGGTypegraphsandSettings, [10](#)  
getSAGSettingsForAStock  
    (getsetSAGSettingsForAStock),  
    [11](#)  
getSAGTypeGraphs  
    (getSAGGTypegraphsandSettings),  
    [10](#)  
getSAGTypeSettings  
    (getSAGGTypegraphsandSettings),  
    [10](#)  
getsetSAGSettingsForAStock, [11](#)  
getSpawningStockBiomassGraph  
    (getStandardAssessmentGraphs),  
    [12](#)  
getSSBHistoricalPerformance  
    (getStandardAssessmentGraphs),  
    [12](#)  
getStandardAssessmentGraphs, [12](#)  
getstock\_recruitment  
    (getStandardAssessmentGraphs),  
    [12](#)  
getStockDownloadData  
    (getStockSourceData), [13](#)  
getStockSourceData, [13](#)  
getStockStatusTable  
    (getStandardAssessmentGraphs),  
    [12](#)  
getStockStatusValues, [14](#)  
getSummaryTable, [2](#), [6–9](#), [15](#)  
getTokenExpiration, [16](#)  
getYSBRSummaryTable, [17](#)  
getYSSB (getStandardAssessmentGraphs),  
    [12](#)  
  
icesSAG (icesSAG-package), [2](#)  
icesSAG-package, [2](#)  
  
readSAGxml (convertSAGxml), [3](#)  
  
setSAGSettingForAStock  
    (getsetSAGSettingsForAStock),  
    [11](#)  
stockFishdata, [3](#), [18](#), [21](#)  
stockInfo, [3](#), [18](#), [21](#)  
  
uploadStock, [20](#)