

Package: icesRDBES (via r-universe)

September 27, 2026

Version 0.0.9

Title Tools for the ICES Regional DataBase and Estimation System

Description Tools to support the ICES Regional DataBase and Estimation System (RDBES) <<https://sboxrdbesapi.ices.dk/rdbescatalogue/>> to access, analyse, and process fisheries data collected under the EU Data Collection Framework.

License file LICENSE

URL <https://github.com/ices-tools-prod/icesRDBES>

Encoding UTF-8

LazyData true

Depends R (>= 4.4)

Imports AzureAuth, httr, jsonlite, base64enc

Suggests rstudioapi

RoxygenNote 7.3.3

Roxygen list(old_usage=TRUE)

Config/pak/sysreqs libssl-dev

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

Date/Publication 2026-08-28 12:40:17 UTC

RemoteUrl <https://github.com/ices-tools-prod/icesRDBES>

RemoteRef HEAD

RemoteSha b04b72f2b422bff6536025df1523e5dbf22195df

Contents

icesRDBES-package	2
decode_token	2
rdbes_api	3
rdbes_download_data	3
rdbes_token	4
rdbes_upload_data	5
type_filters	6
use_sboxrdbes	7

Index**8**

icesRDBES-package	<i>Tools for the ICES Regional DataBase and Estimation System</i>
-------------------	---

Description

Tools to support the ICES Regional DataBase and Estimation System (RDBES), to access, analyse, and process fisheries data collected under the EU Data Collection Framework.

Author(s)

Ejaz Muhammad and Colin Millar.

References

ICES Regional DataBase and Estimation System: <https://sboxshiny.ices.dk/rdbescatalogue/>.

See Also

Useful links:

- <https://github.com/ices-tools-prod/icesRDBES>

decode_token	<i>Decode token</i>
--------------	---------------------

Description

Decodes the token to allow inspection of claims

Usage

```
decode_token(token = rdbes_token())
```

Arguments

token	a javascript web token got by running 'rdbes_token()'
-------	---

Value

list of claims

Examples

```
## Not run:
decode_token()

## End(Not run)
```

`rdbes_api`*Get the current API URL based on the environment*

Description

This function returns the API URL based on whether the package is in development or production mode.

Usage

```
rdbes_api(production = getOption("rdbes.production"), type = c("upload",  
  "download"))
```

Arguments

<code>production</code>	Logical. If TRUE, returns the production API URL. If FALSE, returns the development API URL.
<code>type</code>	Character. The type of API URL to return, either "upload" or "download". Defaults to "upload".

Value

A character string representing the API URL.

`rdbes_download_data`*Download RDBES Data*

Description

This function authenticates with Azure, creates an export job, polls for completion, and downloads the resulting ZIP file to the current working directory.

Usage

```
rdbes_download_data(payload, dest_dir = ".",  
  production = getOption("rdbes.production"), verbose = FALSE,  
  url = NULL)
```

Arguments

<code>payload</code>	List. The filter configuration including hierarchies, format, and filters.
<code>dest_dir</code>	Character. Optional. Directory to save the downloaded file. Defaults to the current working directory.
<code>production</code>	Logical. Optional. Whether to use the production API endpoint. Defaults to <code>getOption("rdbes.production")</code> .

verbose	Logical. Optional. Whether to print verbose HTTP request/response details. Defaults to FALSE.
url	Character. Optional. Custom API URL intended for localhost testing.

Details

The payload should be a nested list. Example: `list(hierarchies = list("H1"), format = "SingleCsvFile", csFilters = list(...))`

Value

Character. The path to the downloaded ZIP file.

Examples

```
## Not run:
my_filters <- list(
  dataType = "SL",
  format = "SingleCsvFile",
  hierarchies = list("HSL"),
  slFilters = list(
    slCountry = list("ZW"),
    slYear = list("2024")
  )
)
rdbes_download_data(my_filters)

## End(Not run)
```

rdbes_token	<i>Get a Token to Download RDBES Data</i>
-------------	---

Description

This function authenticates with Azure using the tenant ID, client app ID, and resource specified in the package options to get an access token.

Usage

```
rdbes_token(invisible = TRUE, quiet = TRUE)
```

Arguments

invisible	Logical. If TRUE (default), the token is returned invisibly to avoid printing it in the console.
quiet	Logical. If TRUE (default), suppresses the token inventory check messages.

Value

Character. The access token string.

Examples

```
## Not run:
token <- rdbes_token()
# Inspect claims (out of curiosity)
decode_token(token)

## End(Not run)
```

rdbes_upload_data	<i>Upload RDBES Data</i>
-------------------	--------------------------

Description

This function authenticates with Azure, uploads a file, starts a screening job, polls for completion, and downloads the resulting reordered file if available.

Usage

```
rdbes_upload_data(file_path, hierarchy,
  production = getOption("rdbes.production"), verbose = FALSE,
  url = NULL)
```

Arguments

file_path	Character. The path to the file to upload.
hierarchy	Character. The hierarchy type for the upload.
production	Logical. Optional. Whether to use the production API endpoint. Defaults to <code>getOption("rdbes.production")</code> .
verbose	Logical. Optional. Whether to print verbose HTTP request/response details. Defaults to <code>FALSE</code> .
url	Character. Optional. Custom API URL intended for localhost testing.

Value

Character. The path to the downloaded ZIP file.

Examples

```
## Not run:
filename <- system.file("test_files/importtest_HNI.csv", package = "icesRDBES")

result <- rdbes_upload_data(file_path = filename, hierarchy = "HNI")

## End(Not run)
```

type_filters	Type Filters for RDBES Data downloads
--------------	---------------------------------------

Description

Type filters for different RDBES data types to facilitate data downloads.

Usage

```
type_filters
```

Format

A list containing type filters for different RDBES data types:

hierarchies	hierarchy
filters	list of filters

Details

Each data type (e.g., Commercial Sampling, Commercial Landing) has its own set of filters that can be applied when requesting data from the RDBES API. These filters are pre-configured for the year 2024 and the country code "ZW" (Zimbabwe) as an example. Users can modify these filters as needed before making their API requests.

Examples

```
type_filters[["CS"]]
```

`use_sboxrdbes`*Set development mode for icesRDBES*

Description

This function allows users to switch between development and production modes for the ices-RDBES package. In development mode, the package will use the sboxrdbes API endpoint (sandbox environment) for testing and development purposes, while in production mode, it will use the production API endpoint.

Usage

```
use_sboxrdbes(flag = TRUE)
```

Arguments

<code>flag</code>	Logical. If TRUE, sets the package to sandbox mode. If FALSE, sets it to production mode. Default is TRUE.
-------------------	--

Details

When in sandbox mode, the package will use the API endpoint specified in the ‘rdbes.api_sandbox_url’ option. When in production mode, it will use the endpoint specified in the ‘rdbes.api_prod_url’ option. This allows for easy switching between testing and live environments without needing to change code.

Examples

```
## Not run:  
# Set to sandbox mode  
use_sboxrdbes(TRUE)  
rdbes_api() # Should return the sandbox API URL  
  
# Set to production mode  
use_sboxrdbes(FALSE)  
rdbes_api() # Should return the production API URL  
  
## End(Not run)
```

Index

`decode_token`, [2](#)

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

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

`rdbes_api`, [3](#)

`rdbes_download_data`, [3](#)

`rdbes_token`, [4](#)

`rdbes_upload_data`, [5](#)

`type_filters`, [6](#)

`use_sboxrdbes`, [7](#)