

Package ‘icesConnect’

July 22, 2025

Title Provides User Tokens for Access to ICES Web Services

Version 1.1.4

Description Provides user tokens for ICES web services that require authentication and authorization. Web services covered by this package are ICES VMS database, the ICES DATSU web services, and the ICES SharePoint site <<https://www.ices.dk/data/tools/Pages/WebServices.aspx>>.

Imports httr, base64enc, jsonlite, askpass

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

URL <https://www.ices.dk/data/tools/Pages/WebServices.aspx>,
<https://github.com/ices-tools-prod/icesConnect>

BugReports <https://github.com/ices-tools-prod/icesConnect/issues>

Language en-GB

NeedsCompilation no

Author Colin Millar [aut, cre]

Maintainer Colin Millar <colin.millar@ices.dk>

Repository CRAN

Date/Publication 2025-04-30 17:30:01 UTC

Contents

clear_token	2
clear_username	2
decode_token	3
ices_get	4
ices_token	6
Index	8

clear_token	<i>clear_token</i>
-------------	--------------------

Description

Deletes the token from the key ring.

Usage

```
clear_token(username = get_username())
```

Arguments

username	the ices username that you require a token for, Default: <code>getOption("ices.username")</code>
----------	--

Value

no return value

Examples

```
## Not run:  
clear_token()  
  
## End(Not run)
```

clear_username	<i>get_username</i>
----------------	---------------------

Description

Get your ICES username from the options.

Set the ICES user or logon name that will be used for requests

Usage

```
clear_username()  
  
get_username()  
  
set_username(username)
```

Arguments

username	the ices username you use to login to SharePoint for example.
----------	---

Value

character scalar, the username
invisible value: the previous username that was set

Examples

```
## Not run:  
get_username()  
  
## End(Not run)  
  
## Not run:  
set_username("myusername")  
  
## End(Not run)
```

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

Description

Decodes the token to allow inspection of claims
Return the expiration date of the token
What user is this token for

Usage

```
decode_token(jwt = ices_token(), formatted = TRUE)  
  
token_expiration(jwt = ices_token())  
  
token_user(jwt = ices_token())
```

Arguments

jwt a javascript web token got by running ices_token()
formatted should formatted output be returned

Value

list of claims
POSIXct date time
string, user name

See Also[username](#)[username](#)[username](#)**Examples**

```
## Not run:
decode_token()

## End(Not run)

## Not run:
token_expiration()

## End(Not run)

## Not run:
token_user()

## End(Not run)
```

*ices_get**ices_request_jwt*

Description

Perform a get request to a url using a javascript web token for authentication

Usage

```
ices_get(
  url,
  retry = TRUE,
  quiet = FALSE,
  verbose = FALSE,
  content = TRUE,
  use_token = FALSE
)
```

```
ices_get_jwt(
  url,
  username = NULL,
  retry = TRUE,
  quiet = FALSE,
  verbose = FALSE,
```

```
    jwt = NULL
)

ices_patch_jwt(
  url,
  body = list(),
  encode = "json",
  username = NULL,
  retry = TRUE,
  quiet = FALSE,
  verbose = FALSE,
  jwt = NULL
)

ices_post(
  url,
  body = list(),
  retry = TRUE,
  verbose = FALSE,
  content = TRUE,
  use_token = TRUE,
  encode = "multipart"
)

ices_post_jwt(
  url,
  body = list(),
  encode = "json",
  username = NULL,
  retry = TRUE,
  quiet = FALSE,
  verbose = FALSE,
  jwt = NULL
)

ices_request_jwt(
  type = c("GET", "POST", "PATCH"),
  url,
  body = NULL,
  encode = c("multipart", "form", "json", "raw"),
  username = NULL,
  retry = TRUE,
  quiet = FALSE,
  verbose = FALSE,
  jwt = NULL
)
```

Arguments

url	the url to make the request to
retry	if initial request fails, should a retry be made, Default: TRUE
quiet	suppress informative messages to the console
verbose	should the http request return verbose output
content	logical, if TRUE the content of the response is returned, if FALSE the http response object is returned
use_token	logical, if TRUE the token is used, if FALSE the token is not used
username	the ices username to perform the request as, Default: <code>getOption("ices.username")</code>
jwt	option to supply a token bypassing the <code>ices_token()</code> route, if "" no token is used
body	optional body of request
encode	If the body is a named list, how should it be encoded?
type	the type of request to make, GET, POST or PATCH

Value

http response object

See Also

[ices_token GET](#), [add_headers](#), [stop_for_status](#), [status_code](#)

Examples

```
## Not run:
# this function is interactive and asks the user for a password
ices_request_jwt("GET", "https://taf.ices.dk/vms/api/gearwidths")
ices_get_jwt("https://taf.ices.dk/vms/api/gearwidths")

the top level function
ices_get("https://taf.ices.dk/vms/api/gearwidths", content = FALSE, use_token = TRUE)
ices_get("https://taf.ices.dk/vms/api/gearwidths", content = TRUE)

## End(Not run)
```

ices_token	<i>ices_token</i>
------------	-------------------

Description

Stores a user token in the system keyring

Usage

```
ices_token(username = NULL, password = NULL, refresh = FALSE, ...)
```

Arguments

username	the ices username that you require a token for, Default: NULL, which results in <code>getOption("ices.username")</code>
password	the ices username that you require a token for, Default: NULL, which results in a dialogue box request for the password
refresh	should the token be refreshed to update new claims for example
...	not yet used

Value

character scalar, the token

Examples

```
## Not run:  
ices_token()  
  
## End(Not run)
```

Index

`add_headers`, [6](#)

`clear_token`, [2](#)

`clear_username`, [2](#)

`decode_token`, [3](#)

`GET`, [6](#)

`get_username (clear_username)`, [2](#)

`ices_get`, [4](#)

`ices_get_jwt (ices_get)`, [4](#)

`ices_patch_jwt (ices_get)`, [4](#)

`ices_post (ices_get)`, [4](#)

`ices_post_jwt (ices_get)`, [4](#)

`ices_request_jwt (ices_get)`, [4](#)

`ices_token`, [6](#), [6](#)

`set_username (clear_username)`, [2](#)

`status_code`, [6](#)

`stop_for_status`, [6](#)

`token_expiration (decode_token)`, [3](#)

`token_user (decode_token)`, [3](#)

`username`, [4](#)