

Package ‘icesConnect’

October 20, 2026

Title Provides User Tokens for Access to ICES Web Services

Version 1.2.0

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

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

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 2026-10-20 20:20:02 UTC

Contents

decode_token	2
ices_get	3
ices_token	5
set_token	6
set_username	7
Index	8

`decode_token`*Decode token*

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, or an empty list if the token is empty or can not be decoded

POSIXct date time

string, user

Examples

```
## Not run:
decode_token()

## End(Not run)

## Not run:
token_expiration()

## End(Not run)

## Not run:
token_user()

## End(Not run)
```

ices_get	<i>ices_request_jwt</i>
----------	-------------------------

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,  
    jwt = NULL  
)
```

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

```
ices_patch_jwt(  
    url,  
    body = list(),  
    encode = "json",  
    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",  
    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"),
  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
jwt	option to supply a token bypassing the ices_token() 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, or NULL if the request could not be made, for example if there is no internet connection or the token is not valid. For ices_get() and ices_post() with content = TRUE, the parsed content of the response, or NULL if the request was not successful.

See Also

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

Examples

```

## Not run:
# requests using a token, set the token first using set_token()
ices_request_jwt("GET", "https://data.ices.dk/vms/api/gearwidths")
ices_get_jwt("https://data.ices.dk/vms/api/gearwidths")

# the top level function

```

```
ices_get("https://data.ices.dk/vms/api/gearwidths", content = FALSE, use_token = TRUE)
ices_get("https://data.ices.dk/vms/api/gearwidths", content = TRUE)

## End(Not run)
```

ices_token

ices_token

Description

Returns the user token stored by `set_token()`, if it is present and has not expired.

Usage

```
ices_token(...)
```

Arguments

... not used, only here for back compatibility

Details

Tokens are created at <https://data.ices.dk/token>. Log in, then copy the R code shown on the page into your R session to store the token. Tokens can not be refreshed from R: when a token expires, get a new one from the same page.

Value

character scalar, the token, or "" if there is no valid token

See Also

`set_token()` to store a token.

Examples

```
## Not run:
ices_token()

## End(Not run)
```

`set_token`*Set your ICES token*

Description

Stores a user token for use in requests to ICES web services that require authentication.

Usage

```
set_token(token)
```

Arguments

`token` character scalar, the token copied from <https://data.ices.dk/token>

Details

Tokens are created at <https://data.ices.dk/token>. Log in, then copy the R code shown on the page into your R session. Tokens can not be refreshed from R: when a token expires, get a new one from the same page.

The token is stored in the user configuration directory given by `tools::R_user_dir("icesConnect", "config")`, replacing any token stored previously.

Value

invisible logical, TRUE if the token was stored

See Also

[ices_token\(\)](#) to get the stored token, and [decode_token\(\)](#) to see who the token is for and when it expires.

Examples

```
## Not run:
set_token("<token copied from https://data.ices.dk/token>")

## End(Not run)
```

set_username	<i>Set your ICES username</i>
--------------	-------------------------------

Description

Back compatability function that does nothing. A username is no longer needed: tokens are created at <https://data.ices.dk/token> and stored using [set_token\(\)](#).

Usage

```
set_username(...)
```

Arguments

... only here for back compatability, does nothing

Value

invisible NULL

Index

`add_headers`, [4](#)

`decode_token`, [2](#)

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

`GET`, [4](#)

`ices_get`, [3](#)

`ices_get_jwt (ices_get)`, [3](#)

`ices_patch_jwt (ices_get)`, [3](#)

`ices_post (ices_get)`, [3](#)

`ices_post_jwt (ices_get)`, [3](#)

`ices_request_jwt (ices_get)`, [3](#)

`ices_token`, [4](#), [5](#)

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

`set_token`, [4](#), [6](#)

`set_token()`, [5](#), [7](#)

`set_username`, [7](#)

`status_code`, [4](#)

`stop_for_status`, [4](#)

`token_expiration (decode_token)`, [2](#)

`token_user (decode_token)`, [2](#)