

# Package: icesConnect (via r-universe)

September 15, 2024

**Title** Provides User Tokens for Access to ICES Web Services

**Version** 1.0.0

**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.

**Imports** whoami, keyring, httr, base64enc, jsonlite, getPass

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Colin Millar [aut, cre]

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

**Date/Publication** 2022-02-10 19:10:02 UTC

**Repository** <https://colinmillar.r-universe.dev>

**RemoteUrl** <https://github.com/cran/icesConnect>

**RemoteRef** HEAD

**RemoteSha** e969d7e34cea2b8e44c8cb5b7677b1b20f05c2bf

## Contents

clear_token . . . . .	2
decode_token . . . . .	2
ices_get_jwt . . . . .	3
ices_token . . . . .	5
set_username . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

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

---

**Description**

Deletes the token from the key ring.

**Usage**

```
clear_token(username = getOption("ices.username"))
```

**Arguments**

username            the ices username that you require a token for, Default: `getOption("ices.username")`

**Value**

no return value

**See Also**

[username key\\_get](#)

**Examples**

```
## Not run:
clear_token()

## 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 key\\_get](#)  
[username key\\_get](#)  
[username key\\_get](#)

**Examples**

```
## Not run:  
decode_token()  
  
## End(Not run)  
  
## Not run:  
token_expiration()  
  
## End(Not run)  
  
## Not run:  
token_user()  
  
## End(Not run)
```

---

ices\_get\_jwt

*ices\_request\_jwt*

---

**Description**

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

**Usage**

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

```
ices_post_jwt(  
    url,  
    body = list(),  
    encode = "json",  
    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_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**

<code>url</code>	the url to make the request to
<code>username</code>	the ices username to perform the request as, Default: <code>getOption("ices.username")</code>

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
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**

httr 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")

## 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

**See Also**

[username key\\_get](#)

**Examples**

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

---

set\_username

*Set your ICES username*

---

**Description**

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

**Usage**

```
set_username(username)
```

**Arguments**

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

**Details**

To set this permanently, add the line: `options(ices.username = "your ices user name")` to your `~/.Rprofile` file, see, usually `~/.Rprofile`, see [Startup](#) for more information.

**Value**

invisible value: the previous username that was set

**See Also**

[options getOption](#)

**Examples**

```
set_username("myusername")
```

# Index

[add\\_headers](#), [5](#)

[clear\\_token](#), [2](#)

[decode\\_token](#), [2](#)

[GET](#), [5](#)

[getOption](#), [6](#)

[ices\\_get\\_jwt](#), [3](#)

[ices\\_patch\\_jwt \(ices\\_get\\_jwt\)](#), [3](#)

[ices\\_post\\_jwt \(ices\\_get\\_jwt\)](#), [3](#)

[ices\\_request\\_jwt \(ices\\_get\\_jwt\)](#), [3](#)

[ices\\_token](#), [5](#), [5](#)

[key\\_get](#), [2](#), [3](#), [6](#)

[options](#), [6](#)

[set\\_username](#), [6](#)

[Startup](#), [6](#)

[status\\_code](#), [5](#)

[stop\\_for\\_status](#), [5](#)

[token\\_expiration \(decode\\_token\)](#), [2](#)

[token\\_user \(decode\\_token\)](#), [2](#)

[username](#), [2](#), [3](#), [6](#)