

Package: tapLock (via r-universe)

October 20, 2026

Title Seamless Single Sign-on for 'shiny'

Version 0.2.0

Description Swift and seamless Single Sign-On (SSO) integration.

Designed for effortless compatibility with popular Single Sign-On providers like Google and Microsoft, it streamlines authentication, enhancing both user experience and application security. Elevate your 'shiny' applications for a simplified, unified, and secure authentication process.

License MIT + file LICENSE

URL <https://github.com/ixpantia/tapLock>

BugReports <https://github.com/ixpantia/tapLock/issues>

Depends R (>= 4.2.0)

Imports glue, promises, stringr, shiny, later, methods, tower (>= 0.2.0), rlang

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/rextendr/version 0.4.2

SystemRequirements Cargo (Rust's package manager), rustc, OpenSSL >= 1.0.2

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev zlib1g-dev libclang-dev

Repository <https://ixpantia.r-universe.dev>

Date/Publication 2025-11-24 21:06:21 UTC

RemoteUrl <https://github.com/ixpantia/taplock>

RemoteRef HEAD

RemoteSha 7879e536c1451e722782e6fd13a7901311629f03

Contents

add_auth_layers	2
expires_at	3
expires_in	3
hello_world	4
is_expired	4
is_valid	4
new_entra_id_config	5
new_google_config	5
new_keycloak_config	6
new_openid_config	7
print.access_token	7
token	8
Index	9

add_auth_layers	<i>Add authentication middle ware to a 'tower' object</i>
-----------------	---

Description

Attaches the necessary authentication layers to a 'tower' object. This will secure any layer added after.

Usage

```
add_auth_layers(tower, config)
```

Arguments

- tower A 'tower' object from the package 'tower'
- config An 'openid_config' object

Value

A modified 'tower' object with authentication layers

expires_at	<i>Get the expiration date and time of an access token</i>
------------	--

Description

Gets the expiration date and time of an access token

Usage

```
expires_at(token)
```

Arguments

token	An access_token object
-------	------------------------

Value

A POSIXct object containing the date and time the token expires

expires_in	<i>Get the expiration time of an access token</i>
------------	---

Description

Gets the expiration time of an access token

Usage

```
expires_in(token)
```

Arguments

token	An access_token object
-------	------------------------

Value

A duration object containing the time until the token expires

hello_world	<i>Return string "Hello world!" to R.</i>
-------------	---

Description

Return string "Hello world!" to R.

Usage

```
hello_world()
```

is_expired	<i>Check if an access token is expired</i>
------------	--

Description

Checks if an access token is expired

Usage

```
is_expired(token)
```

Arguments

token	An access_token object
-------	------------------------

Value

A boolean indicating if the token is expired

is_valid	<i>Check if an access token is valid</i>
----------	--

Description

Checks if an access token is valid by checking if it is expired

Usage

```
is_valid(token)
```

Arguments

token	An access_token object
-------	------------------------

Value

A boolean indicating if the token is valid

new_entra_id_config	Create a new entra_id_config object
---------------------	-------------------------------------

Description

Creates a new entra_id_config object

Usage

```
new_entra_id_config(  
    tenant_id,  
    client_id,  
    client_secret,  
    app_url,  
    use_refresh_token = TRUE  
)
```

Arguments

tenant_id	The tenant ID for the app
client_id	The client ID for the app
client_secret	The client secret for the app
app_url	The URL for the app
use_refresh_token	Enable the use of refresh tokens

Value

An entra_id_config object

new_google_config	Create a new google_config object
-------------------	-----------------------------------

Description

Creates a new google_config object

Usage

```
new_google_config(client_id, client_secret, app_url, use_refresh_token = TRUE)
```

Arguments

- client_id The client ID for the app
- client_secret The client secret for the app
- app_url The URL for the app
- use_refresh_token Enable the use of refresh tokens

Value

A google_config object

new_keycloak_config	Create a new keycloak_config object
---------------------	-------------------------------------

Description

Creates a new keycloak_config object

Usage

```
new_keycloak_config(  
  base_url,  
  realm,  
  client_id,  
  client_secret,  
  app_url,  
  use_refresh_token = TRUE  
)
```

Arguments

- base_url The base URL for the Keycloak instance
- realm The realm for the app
- client_id The client ID for the app
- client_secret The client secret for the app
- app_url The URL for the app
- use_refresh_token Enable the use of refresh tokens

Value

A keycloak_config object

new_openid_config	<i>New openid configuration</i>
-------------------	---------------------------------

Description

Creates a new openid configuration object for the given provider. You can use this function or the individual provider functions.

Usage

```
new_openid_config(provider, app_url, ...)
```

Arguments

provider	The openid provider to use
app_url	The URL of the application (used to build redirect, login, and logout URLs)
...	Additional arguments passed to the provider's configuration. This depends on the provider. The "google" provider accepts the following arguments: • client_id • client_secret The "entra_id" provider accepts the following arguments: • client_id • client_secret • tenant_id

Value

An openid_config object

print.access_token	<i>Print an access token</i>
--------------------	------------------------------

Description

Prints an access token's expiration date

Usage

```
## S3 method for class 'access_token'  
print(x, ...)
```

Arguments

x	An access_token object
...	Ignored

Value

No return value, called for side effects

token	<i>Get the access token</i>
-------	-----------------------------

Description

Gets the access token from the session to be used for internal logic.

Usage

```
token(session = shiny::getDefaultReactiveDomain())
```

Arguments

session	A Shiny session
---------	-----------------

Value

An access_token object

Index

`add_auth_layers`, [2](#)

`expires_at`, [3](#)

`expires_in`, [3](#)

`hello_world`, [4](#)

`is_expired`, [4](#)

`is_valid`, [4](#)

`new_entra_id_config`, [5](#)

`new_google_config`, [5](#)

`new_keycloak_config`, [6](#)

`new_openid_config`, [7](#)

`print.access_token`, [7](#)

`token`, [8](#)