Package ‘tapLock’

October 28, 2024

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`

Imports curl, glue, httr2, jose, lubridate, promises, purrr, stats, stringr, shiny, tower (>= 0.2.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author ixpantia, SRL [cph],
Andres Quintero [aut, cre]

Maintainer Andres Quintero <`andres@ixpantia.com`>

Repository CRAN

Date/Publication 2024-10-28 11:50:08 UTC

Contents

`add_auth_layers` *Add authentication middle ware to a 'tower' object*

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` *Get the expiration date and time of an access token*

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` *Get the expiration time of an access token*

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

---

`get_token_field` *Get the issued at time of an access token*

Description

Gets the issued at time of an access token

Usage

```
get_token_field(token, field)
```

Arguments

`token`	An access_token object
`field`	The field to get from the token

Value

A POSIXct object containing the date and time the token was issued

is_expired *Check if an access token is expired*

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 *Check if an access token is valid*

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_auth0_config` *Create a new auth0_config object*

Description

Creates a new auth0_config object

Usage

```
new_auth0_config(client_id, client_secret, auth0_domain, app_url)
```

Arguments

`client_id`	The client ID for the app
`client_secret`	The client secret for the app
`auth0_domain`	The domain for the Auth0 tenant
`app_url`	The URL for the app

Value

An auth0_config object

`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)
```

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

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)
```

Arguments

`client_id` The client ID for the app

`client_secret` The client secret for the app

`app_url` The URL for the app

Value

A google_config object

`new_openid_config` *New openid configuration*

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`

The `"auth0"` provider accepts the following arguments:

- `client_id`
- `client_secret`
- `auth0_domain`

Value

An openid_config object

`print.access_token` *Print an access token*

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 *Get the access token*

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