

# Package ‘auth0’

July 22, 2025

**Type** Package

**Title** Authentication in Shiny with Auth0

**Version** 0.2.3

**Description** Uses Auth0 API (see <<https://auth0.com>> for more information) to use a simple authentication system. It provides tools to log in and out a shiny application using social networks or a list of e-mails.

**BugReports** <https://github.com/curso-r/auth0/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** httr, shiny, yaml, utils, shinyjs

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**URL** <https://curso-r.github.io/auth0/>

**NeedsCompilation** no

**Author** Julio Trecenti [cre],  
Daniel Falbel [aut],  
José Jesus [ctb],  
Dean Attali [ctb],  
C Lente [ctb]

**Maintainer** Julio Trecenti <[julio.trecenti@gmail.com](mailto:julio.trecenti@gmail.com)>

**Repository** CRAN

**Date/Publication** 2023-03-26 21:50:05 UTC

## Contents

auth0_config . . . . .	2
auth0_find_config_file . . . . .	2
auth0_info . . . . .	3

auth0_logout_url . . . . .	3
logout . . . . .	4
logoutButton . . . . .	5
logout_url . . . . .	6
shinyAppAuth0 . . . . .	6
shinyAuth0App . . . . .	7
ui-server . . . . .	7
use_auth0 . . . . .	8

## Index 10

---

auth0_config	<i>Parse _auth0.yml file.</i>
--------------	-------------------------------

---

### Description

Validates and creates a list of useful information from the `_auth0.yml` file.

### Usage

```
auth0_config(config_file)
```

### Arguments

config_file	path to the <code>_auth0.yml</code> file. If not informed, will try to find the file using <a href="#">auth0_find_config_file</a> .
-------------	---

### Value

List containing all the information from the `_auth0.yml` file.

---

auth0_find_config_file	<i>Find the configuration file.</i>
------------------------	-------------------------------------

---

### Description

Tries to find the path to the `_auth0.yml` file. First, it tries to get this info from `options(auth0_config_file = )`. If this option is NULL (the default) it tries to find the `_auth0.yml` within the working directory. If the file does not exist, it raises an error.

### Usage

```
auth0_find_config_file()
```

**Value**

Character vector of length one containing the path of the `_auth0.yml` file.

**See Also**

[use\\_auth0](#).

---

auth0\_info

*Information used to connect to Auth0.*

---

**Description**

Creates a list containing all the important information to connect to Auth0 service's API.

**Usage**

```
auth0_info(config)
```

**Arguments**

`config` path to the `_auth0.yml` file or the object returned by [auth0\\_config](#). If not informed, will try to find the file using [auth0\\_find\\_config\\_file](#).

**Value**

A list containing scope, state, keys, OAuth2.0 app, endpoints, audience and remote URL. For compatibility reasons, `remote_url` can be either a parameter in the root yml or inside a `shiny_config` parameter.

**See Also**

[use\\_auth0](#) to create an `_auth0.yml` template.

---

auth0\_logout\_url

*Generate logout URL*

---

**Description**

`auth0_logout_url()` is defunct as of `auth0` 0.1.2 in order to simplify the user experience with the [logoutButton\(\)](#) function.

**Usage**

```
auth0_logout_url(config_file = NULL, redirect_js = TRUE)
```

**Arguments**

`config_file` Path to YAML configuration file.  
`redirect_js` include javascript code to redirect page? Defaults to TRUE.

**Examples**

```
# simple UI with action button
# AFTER auth0 0.1.2

if (interactive()) {
  library(shiny)
  library(auth0)

  ui <- fluidPage(logoutButton())
  server <- function(input, output, session) {}
  config_file <- system.file("simple/_auth0.yml", package = "auth0")
  shinyAppAuth0(ui, server, config_file)

# simple UI with action button
# BEFORE auth0 0.1.2

library(shiny)
library(auth0)
library(shinyjs)

# note that you must include shinyjs::useShinyjs() for this to work
ui <- fluidPage(shinyjs::useShinyjs(), actionButton("logout_auth0", "Logout"))

# server with one observer that logsouts
server <- function(input, output, session) {
  observeEvent(input$logout_auth0, {
    # javascript code redirecting to correct url
    js <- auth0_logout_url()
    shinyjs::runjs(js)
  })
}
shinyAuth0App(ui, server, config_file)
}
```

---

`logout`*Log out of an auth0 app*

---

**Description**

Log the current user out of an auth0 shiny app.

**Usage**

```
logout()
```

**Details**

You can also use a different configuration file by setting the `auth0_config_file` option with: `options(auth0_config_file = "path/to/file.yaml")`.

**See Also**

[logoutButton](#), [logout\\_url](#)

---

logoutButton	<i>Create a button to log out</i>
--------------	-----------------------------------

---

**Description**

A `logoutButton` is an [actionButton](#) that is meant to be used to log out of an auth0 Shiny app.

**Usage**

```
logoutButton(label = "Log out", ..., id = "._auth0logout_")
```

**Arguments**

<code>label</code>	The label on the button.
<code>...</code>	Named attributes to apply to the button.
<code>id</code>	An ID for the button. If you only have one logout button in your app, you do not need to explicitly provide an ID. If you have more than one logout button, you need to provide a unique ID to each button. When you create a button with a non-default ID, you must create an observer that listens to a click on this button and logs out of the app with a call to <a href="#">logout</a> .

**See Also**

[logout](#), [logout\\_url](#)

**Examples**

```
if (interactive()) {
  ui <- fluidPage(
    logoutButton(),
    logoutButton(label = "Another logout button", id = "logout2")
  )
  server <- function(input, output, session) {
    observeEvent(input$logout2, {
      logout()
    })
  }
}
```

```

    }
    shinyAuth0App(ui, server)
  }

```

---

logout_url	<i>Generate a logout URL</i>
------------	------------------------------

---

### Description

Generate a URL that will log the user out of the app if visited.

### Usage

```
logout_url()
```

### Details

You can also use a different configuration file by setting the `auth0_config_file` option with: `options(auth0_config_file = "path/to/file.yaml")`.

### Value

URL string to log out.

### See Also

[logoutButton](#), [logout](#)

---

shinyAppAuth0	<i>Create a Shiny app object with Auth0 Authentication</i>
---------------	--

---

### Description

This function modifies `ui` and `server` objects to run using Auth0 authentication.

### Usage

```
shinyAppAuth0(ui, server, config_file = NULL, ...)
```

### Arguments

<code>ui</code>	an ordinary UI object to create shiny apps.
<code>server</code>	an ordinary server object to create shiny apps.
<code>config_file</code>	path to YAML configuration file.
<code>...</code>	Other arguments passed on to <a href="#">shiny::shinyApp()</a> .

**Details**

You can also use a different configuration file by setting the `auth0_config_file` option with: `options(auth0_config_file = "path/to/file.yaml")`.

**Disable auth0 while developing apps**

Sometimes, using auth0 to develop and test apps can be frustrating, because every time the app is started, auth0 requires the user to log-in. To avoid this problem, one can run `options(auth0_disable = TRUE)` to disable auth0 temporarily.

---

shinyAuth0App

*Create a Shiny app object with Auth0 Authentication*


---

**Description**

As of auth0 0.1.2, `shinAuth0App()` has been renamed to [shinyAppAuth0\(\)](#) for consistency.

**Usage**

```
shinyAuth0App(ui, server, config_file = NULL)
```

**Arguments**

<code>ui</code>	an ordinary UI object to create shiny apps.
<code>server</code>	an ordinary server object to create shiny apps.
<code>config_file</code>	path to YAML configuration file.

---

ui-server

*Modifies ui/server objects to authenticate using Auth0.*


---

**Description**

These functions can be used in a `ui.R/server.R` framework, modifying the shiny objects to authenticate using Auth0 service with no pain.

**Usage**

```
auth0_ui(ui, info)
```

```
auth0_server(server, info)
```

**Arguments**

ui	shiny.tag.list object to generate the user interface.
info	object returned from <a href="#">auth0_info</a> . If not informed, will try to find the <code>_auth0.yml</code> and create it automatically.
server	the shiny server function.

**See Also**

[auth0\\_info](#).

**Examples**

```
# first, create the yml file using use_auth0() function

if (interactive()) {

  # ui.R file
  library(shiny)
  library(auth0)
  auth0_ui(fluidPage(logoutButton()))

  # server.R file
  library(auth0)
  auth0_server(function(input, output, session) {})

  # console
  options(shiny.port = 8080)
  shiny::runApp()
}
```

---

use_auth0	<i>Auth0 configuration file</i>
-----------	---------------------------------

---

**Description**

Create an YAML containing information to connect with Auth0.

**Usage**

```
use_auth0(path = ".", file = "_auth0.yml", overwrite = FALSE)
```

**Arguments**

path	Directory name. Should be the root of the shiny app you want to add this functionality
file	File name. Defaults to <code>_auth0.yml</code> .
overwrite	Will only overwrite existing path if TRUE.



## Details

The YAML configuration file has required parameters and extra parameters.

The required parameters are:

- `auth0_config` is a list containing at least:
  - `api_url`: Your account at Auth0 (e.g. <https://jonhdoe.auth0.com>). It is the "Domain" in Auth0 application settings.
  - `credentials`: Your credentials to access Auth0 API, including
    - \* `key`: the Client ID in Auth0 application settings.
    - \* `secret`: the Client Secret in Auth0 application settings.

The extra parameters are:

- `remote_url`: If you are using Shiny-Server or ShinyApps IO service.
- `scope`: The information that Auth0 app will access. Defaults to "openid profile".
- `request`: Endpoint to request a token. Defaults to "oauth/token"
- `access`: Endpoint to access. Defaults to "oauth/token"

# Index

actionButton, [5](#)  
auth0\_config, [2](#), [3](#)  
auth0\_find\_config\_file, [2](#), [2](#), [3](#)  
auth0\_info, [3](#), [8](#)  
auth0\_logout\_url, [3](#)  
auth0\_server (ui-server), [7](#)  
auth0\_ui (ui-server), [7](#)

logout, [4](#), [5](#), [6](#)  
logout\_url, [5](#), [6](#)  
logoutButton, [5](#), [5](#), [6](#)  
logoutButton(), [3](#)

shiny::shinyApp(), [6](#)  
shinyAppAuth0, [6](#)  
shinyAppAuth0(), [7](#)  
shinyAuth0App, [7](#)

ui-server, [7](#)  
use\_auth0, [3](#), [8](#)