Package 'cookiemonster'

November 30, 2025

Title Your Friendly Solution to Managing Browser Cookies

Version 0.1.0

Description A convenient tool to store and format browser cookies and use them in 'HTTP' requests (for example, through 'httr2', 'httr' or 'curl').

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Suggests curl, DBI, dplyr, httr, httr2, jsonlite, knitr, rmarkdown, RSQLite, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

Imports cli, openssl, rappdirs, rlang, stringi, tibble, urltools, vctrs

Config/testthat/edition 3

Language en-GB

URL https://github.com/JBGruber/cookiemonster

BugReports https://github.com/JBGruber/cookiemonster/issues

NeedsCompilation no

Author Johannes B. Gruber [aut, cre] (ORCID: <https://orcid.org/0000-0001-9177-1772>)

Maintainer Johannes B. Gruber <JohannesB.Gruber@gmail.com>

Repository CRAN

Date/Publication 2025-11-30 09:10:02 UTC

Contents

`add_cookies` *Add Cookies to the Browser*

Description

This function allows you to add browser cookies to the cookie storage. It can work with either a cookie file or a direct cookie string (e.g., copied from a CURL call). But remember, just like in real life, you can't have your cookie and eat it too - pick only one!

Usage

```
add_cookies(cookiefile, cookiestring, session, domain = NULL, confirm = FALSE)
```

Arguments

`cookiefile`	A character string indicating the path to the cookie file.
`cookiestring`	A character string representing the cookie in string format.
`session`	A live html session created with read_html_live (only version >= 1.0.4).
`domain`	An optional parameter that specifies the host/domain. It's only used when `cookiestring` is provided.
`confirm`	If `TRUE`, you confirm to write the cookie jar to disk (if it does not exist yet) without seeing the interactive menu.

Value

No explicit return. Instead, this function stores the cookies using the `store_cookies` function.

Note

You can't provide both a cookiefile and a cookiestring at the same time. That's like trying to dunk two cookies in a tiny cup of milk!

Your cookies are saved in an encrypted file. See encrypt_vec for more info.

See Also

`store_cookies`

Examples

```
# to conform with CRAN policies, examples use a temporary location. Do not use
# the options like this, except you want your cookies gone when closing R.
options(cookie_dir = tempdir())

# Using a cookie file:
# to conform with CRAN policies, examples use a temporary location. Do not use
# the options like this, except you want your cookies gone when closing R.
add_cookies(cookiefile = system.file("extdata", "cookies.txt", package = "cookiemonster"))

# Using a cookie string:
add_cookies(cookiestring = "username=johndoe; password=secret", domain = "www.example.com")
```

`default_jar` *Get the default cookie storage directory (jar)*

Description

This function returns the default directory (jar) for storing cookies. Users can set their own cookie storage location by using `options(cookie_dir = "your/directory/here")`. If no custom directory is specified, the default directory used by the `rappdirs` package will be returned.

Usage

```
default_jar()
```

Value

A string representing the path to the default cookie storage directory (jar).

Examples

```
# Get the default jar
default_jar()

# Set a custom cookie storage directory
options(cookie_dir = "/path/to/your/cookie/directory")
# Get the custom cookie directory
default_jar()

# revert to the package default
options(cookie_dir = rappdirs::user_cache_dir("r_cookies"))
```

delete_cookies *Delete Cookies*

Description

Delete Cookies

Usage

```
delete_cookies(
  domain,
  key = "",
  jar = default_jar(),
  fixed = FALSE,
  ask = TRUE
)
```

Arguments

domain	The domain for which the cookies should be deleted.
key	An optional filter to retrieve only certain cookies by matching certain keys/names. Accepts regular expression depending on the value of `fixed`.
jar	A character string of the path to the cookie jar (the default is to use `default_jar()` to get a suitable directory).
fixed	If `TRUE`, `domain` and `key` are matched as is. If either `domain` or `key`, only those values are treated as is.
ask	A logical value indicating whether the user should be asked to confirm the deletion.

Value

Nothing. Called to remove cookies from jar.

Examples

```
# to conform with CRAN policies, examples use a temporary location. Do not use
# the options like this, except you want your cookies gone when closing R.
options(cookie_dir = tempdir())

add_cookies(cookiefile = system.file("extdata", "cookies.txt", package = "cookiemonster"))
delete_cookies("example.com", ask = FALSE)
```

encrypt_vec *Encrypts/Decrypts a vector*

Description

Used internally to encrypt/decrypt the value column of your cookie jar.

Usage

```
encrypt_vec(vec)

decrypt_vec(vec)
```

Arguments

vec	A vector of values to encrypt

Details

If you save valuable cookies, for example login information, you should encrypt them with a personalised password. This can be set with, e.g., `Sys.setenv("COOKIE_KEY" = "megageheim")` or in an .Renviron file.

Value

list of encrypted elements (for `encrypt_vec`); vector of decrypted elements (for `encrypt_vec`).

Examples

```
enc <- encrypt_vec(c("foo", "bar"))
decrypt_vec(enc)
```

get_cookies *Retrieve cookies from a jar*

Description

Imagine you're reaching into a magical jar overflowing with those scrumptious digital delights from websites you've visited. The flavour? Up to you! Just select your desired output format.

Usage

```
get_cookies(
  domain,
  key = "",
  jar = default_jar(),
  as = c("data.frame", "string", "vector", "list"),
  fixed = FALSE
)
```

Arguments

domain	A character string of the domain to retrieve cookies for. Accepts regular expression depending on the value of fixed.
key	An optional filter to retrieve only certain cookies by matching certain keys/names. Accepts regular expression depending on the value of fixed.
jar	A character string of the path to the cookie jar (the default is to use default_jar() to get a suitable directory).
as	A character string of the type of output to return.
fixed	If TRUE, domain and key are matched as is. If either domain or key, only those values are treated as is.

Details

The function returns cookies in one of three formats:

- **data.frame:** is how cookies are stored internally and can be used for manual inspection.
- **string:** is used by curl and httr2.
- **vector:** is used by httr.
- **list:** is used by chromote.

See vignette("cookies", "cookiemonster") for how to use cookies with these packages.

Value

Depending on the value of as, returns either a data frame, a character string, or a named vector.

Note

Your cookies are saved in an encrypted file. See encrypt_vec for more info.

See Also

add_cookies

Examples

```
# to conform with CRAN policies, examples use a temporary location. Do not use the options like this
options(cookie_dir = tempdir())

# put some cookies in the jar
add_cookies(cookiestring = "chococookie=delicious", domain = "example.com")
# Reach into your cookie jar and enjoy!
get_cookies("example.com")
# put different cookies into the jar (overwrites previous)
add_cookies(cookiestring = "oatmeal=delicious; peanutbutter=delicious", domain = "example.com")
add_cookies(cookiestring = "snickerdoodle=delicious", domain = "another.example.com")
# only get cookies for example.com, not another.example.com
get_cookies("^example.com")
# only get some cookies from example.com
get_cookies(domain = "^example.com", key = "peanut")
```

---

get_cookies_from_browser

Get Cookies from Browser

Description

Retrieves all cookies from a Browser (currently works with Firefox only).

Usage

```
get_cookies_from_browser(browser = "Firefox")
```

Arguments

browser	A character string specifying the browser to get cookies from. Defaults to "Firefox".

Value

A tibble containing the cookies from the specified browser.

Examples

```
## Not run:
get_cookies_from_browser("Firefox")

## End(Not run)
```

req_cookiemonster_set *Set cookies from the cookiejar to a httr2 request*

Description

Adds all cookies for the domain from the jar to a httr2 request.

Usage

```
req_cookiemonster_set(req, jar = default_jar(), ...)
```

Arguments

req	A request object.
jar	A character string of the path to the cookie jar (the default is to use default_jar() to get a suitable directory).
...	not currently used.

Value

A request object (with cookies).

Examples

```
library(httr2)
domain <- url_parse(example_url())$hostname
add_cookies(cookiestring = "snicker=doodle; password=secret", domain = domain)
request(example_url()) |>
  req_url_path("/cookies/set") |>
  req_cookiemonster_set() |>
  req_perform() |>
  resp_body_json()
```

store_cookies *Store cookies in a jar*

Description

Store cookies in a jar

Usage

```
store_cookies(cookies, jar = default_jar(), confirm = FALSE)
```

Arguments

cookies	A data frame of cookies
jar	The directory to store the cookies in. Defaults to `default_jar()`.
confirm	If TRUE, you confirm to write the cookie jar to disk (if it does not exist yet) without seeing the interactive menu.

Value

No return value, called to save (encrypted) cookies on disk.

Examples

```
# to conform with CRAN policies, examples use a temporary location. Do not use
# the options like this, except you want your cookies gone when closing R.
options(cookie_dir = tempdir())

if (requireNamespace("curl", quietly = TRUE)) {
  # get cookies from a curl request
  library(curl)
  h <- new_handle()
 resp <- curl_fetch_memory("https://hb.cran.dev/cookies/set?new_cookies=moo", handle = h)
  cookies <- handle_cookies(h)

  # then store them for future use
  store_cookies(cookies)

  # then you can retrieve them and use in future calls
  get_cookies("hb.cran.dev")
}
```

Index