

Package: cookie (via r-universe)

June 13, 2026

Title HTTP Cookies Parser Middleware

Version 1.0.0

Description A cookie is a piece of data sent from a web server to a web client which helps in overcoming the statelessness constraint of the HTTP protocol. This package provides the tools to work with them in the form of a cookie parser middleware function, meant to be attached to a bigger R web application, and utilities to write, sign and unsign a cookie. For more details see the Mozilla Developer Network (MDN) web documentation in <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Cookies>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports secretbase, utils

Suggests curl, httpuv, later, R6, routing, testthat (>= 3.0.0),
yyjsonr

Config/testthat/edition 3

Repository https://juliocollazos64.r-universe.dev

Date/Publication 2026-06-06 04:40:06 UTC

RemoteUrl https://github.com/juliocollazos64/cookie

RemoteRef HEAD

RemoteSha cd48ce7351b4b5c4ad07aa1f252b689dc9271e6d

Contents

cookieParser	2
serialise	2
sign	3
unsign	4

Index	5
--------------	----------

cookieParser	<i>Cookie Parser Middleware</i>
--------------	---------------------------------

Description

Creates a middleware function that parses the Cookie HTTP header and populates req\$cookies. When a secret is provided, signed cookies are verified and exposed on req\$signedCookies.

Usage

```
cookieParser(secret = NULL, options = NULL)
```

Arguments

secret	A character vector or list of strings used to sign and verify cookies. Optional.
options	A function used to decode cookie values. Defaults to <code>utils::URLdecode()</code> .

Value

A middleware function that sets req\$cookies, req\$signedCookies, and req\$secret, then calls forward().

serialise	<i>Serialise a Set-Cookie Header</i>
-----------	--------------------------------------

Description

Serialises a cookie name-value pair into a Set-Cookie header string.

Usage

```
serialise(name, val = NULL, ...)
```

Arguments

name	A string with the cookie name, or a list with \$name and \$value elements (in which case val is ignored).
val	A string. The cookie value.
...	Additional cookie attributes:
encode	A function to encode the cookie value. Defaults to <code>utils::URLencode()</code> .
maxAge	An integer. Number of seconds until the cookie expires.
domain	A string. The cookie domain.
path	A string. The cookie path.
expires	A Date, POSIXct, or POSIXt. The expiry date.

httpOnly Logical. Adds the HttpOnly attribute.
 secure Logical. Adds the Secure attribute.
 partitioned Logical. Adds the Partitioned attribute.
 priority A string: "low", "medium", or "high".
 sameSite A string ("strict", "lax", "none") or logical (TRUE maps to "Strict").

Value

A Set-Cookie header string.

Examples

```

serialise("session", "abc123")
serialise("id", "42", httpOnly = TRUE, secure = TRUE, sameSite = "lax")

```

sign

Sign a Cookie Value

Description

Sign the given val with secret.

Usage

```
sign(val, secret)
```

Arguments

val A string. The cookie value to sign.
 secret The secret key used to generate the signature.

Value

A string of the form "<val>.<signature>".

Examples

```
sign("hello", "tobiiscool")
```

`unsign`*Unsign a Cookie Value*

Description

Verifies the signature of a signed cookie value and returns the original value if valid, or FALSE if the signature does not match.

Usage

```
unsign(input, secret)
```

Arguments

<code>input</code>	A string. A signed cookie value produced by <code>sign()</code> .
<code>secret</code>	The secret key to verify against.

Value

The original unsigned value if verification succeeds, FALSE otherwise.

Examples

```
input <- sign("hello", "tobiiscool")
unsign(input, "tobiiscool")
unsign(input, "luna")
```

Index

`cookieParser`, [2](#)

`serialise`, [2](#)

`sign`, [3](#)

`sign()`, [4](#)

`unsign`, [4](#)

`utils::URLdecode()`, [2](#)

`utils::URLEncode()`, [2](#)