

# Package: glouton (via r-universe)

October 4, 2024

**Type** Package

**Title** 'JS-cookies' for 'Shiny'

**Version** 0.0.0.9000

**Description** Set and get browser cookies in 'Shiny' through  
'JS-cookie'.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** shiny

**Roxygen** list(markdown = TRUE)

**Repository** <https://thinkr-open.r-universe.dev>

**RemoteUrl** <https://github.com/ColinFay/glouton>

**RemoteRef** HEAD

**RemoteSha** f6b466be86800a945f1677022d93aac4da57aeb1

## Contents

Cookie . . . . .	2
fetch_cookies . . . . .	3
use_glouton . . . . .	4
<b>Index</b>	<b>5</b>

---

Cookie

Create a Cookie

---

## Description

Create a cookie object.

## Methods

### Public methods:

- `Cookie$new()`
- `Cookie$get()`
- `Cookie$set()`
- `Cookie$rm()`
- `Cookie$clone()`

### Method `new()`:

*Usage:*

```
Cookie$new(name, session = shiny::getDefaultReactiveDomain())
```

*Arguments:*

`name` The name of the cookie.

`session` A valid Shiny session.

*Details:* Create a Cookie

### Method `get()`:

*Usage:*

```
Cookie$get()
```

*Details:* Get the value of the cookie

### Method `set()`:

*Usage:*

```
Cookie$set(value, expires = 365L, path = "/")
```

*Arguments:*

`value` Value of the cookie

`expires` When the cookie is set to expire, integer indicating number of days.

`path` Path where cookie is visible.

*Details:* Set the value of the cookie

### Method `rm()`:

*Usage:*

```
Cookie$rm()
```

*Details:* Remove the cookie.

**Method** clone(): The objects of this class are cloneable with this method.

*Usage:*

```
Cookie$clone(deep = FALSE)
```

*Arguments:*

deep Whether to make a deep clone.

---

 fetch\_cookies

*Add and fetch browsers cookies*


---

## Description

fetch\_cookies returns all the cookies, while fetch\_cookie search for one cookie in the browser.

## Usage

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

```
fetch_cookie(name, session = shiny::getDefaultReactiveDomain())
```

```
add_cookie(
  name,
  value,
  options = cookie_options(),
  debug = FALSE,
  session = shiny::getDefaultReactiveDomain()
)
```

```
remove_cookie(
  name,
  options = cookie_options(),
  session = shiny::getDefaultReactiveDomain()
)
```

```
cookie_options(
  expires = NULL,
  path = "/",
  domain = NULL,
  secure = FALSE,
  sameSite = "strict"
)
```

## Arguments

session	The session object passed to function given to shinyServer. Default is <a href="#">shiny::getDefaultReactiveD</a>
name	Name of the cookie to set/fetch
value	The value to set for the cookie

options	A list of options returned from <code>cookie_options()</code> . The same options that were passed to <code>add_cookie()</code> must be passed to <code>remove_cookie()</code> .
debug	If TRUE, a message is displayed in the browser console after adding a cookie.
expires	Define when the cookie will be removed. Value must be a numeric, which will be interpreted as days from time of creation or a date. If NULL, the cookie becomes a session cookie.
path	A string indicating the path where the cookie is visible.
domain	A string indicating a valid domain where the cookie should be visible. The cookie will also be visible to all subdomains.
secure	Boolean indicating whether the cookie transmission requires a secure protocol (https).
sameSite	A string allowing to control whether the browser is sending a cookie along with cross-site requests.

---

use_glouton	<i>Add glouton to your shiny app</i>
-------------	--------------------------------------

---

### Description

Add glouton to your shiny app

### Usage

```
use_glouton(online = TRUE)
```

### Arguments

online	TRUE fetch the online version of js-cookie through CDN. FALSE uses the file included in the pckage.
--------	---

# Index

`add_cookie (fetch_cookies), 3`

`Cookie, 2`

`cookie_options (fetch_cookies), 3`

`fetch_cookie (fetch_cookies), 3`

`fetch_cookies, 3`

`remove_cookie (fetch_cookies), 3`

`shiny::getDefaultReactiveDomain(), 3`

`use_glouton, 4`