

Discover Packages > Standard library > net > http > cookiejar 

cookiejar


package

standard library

Version: [go1.20.1](#) **Latest** | Published: Feb 14, 2023 | License: [BSD-3-Clause](#) | Imports: [11](#) |

Imported by: [6,016](#)

Details

 Valid [go.mod](#) file  Redistributable license  Tagged version  Stable version 

[Learn more](#)

Repository

cs.opensource.google/go/go

Links

 [Report a Vulnerability](#)

 Documentation 

<> Documentation

Overview

Package cookiejar implements an in-memory [RFC 6265](#)-compliant `http.CookieJar`.

Index

[type Jar](#)

```
func New(o *Options) (*Jar, error)
```

```
func (j *Jar) Cookies(u *url.URL) (cookies []*http.Cookie)
```

```
func (j *Jar) SetCookies(u *url.URL, cookies []*http.Cookie)
```

[type Options](#)

[type PublicSuffixList](#)

Examples

[New](#)

Constants

This section is empty.

Variables

This section is empty.

Functions

This section is empty.

Types

type Jar

```
type Jar struct {  
    // contains filtered or unexported fields  
}
```

Jar implements the `http.CookieJar` interface from the `net/http` package.

func New

```
func New(o *Options) (*Jar, error)
```

New returns a new cookie jar. A nil `*Options` is equivalent to a zero `Options`.

► [Example](#)

func (*Jar) Cookies

```
func (j *Jar) Cookies(u *url.URL) (cookies []*http.Cookie)
```

Cookies implements the `Cookies` method of the `http.CookieJar` interface.

It returns an empty slice if the URL's scheme is not HTTP or HTTPS.

func (*Jar) SetCookies

```
func (j *Jar) SetCookies(u *url.URL, cookies []*http.Cookie)
```

SetCookies implements the `SetCookies` method of the `http.CookieJar` interface.

It does nothing if the URL's scheme is not HTTP or HTTPS.

type Options

```
type Options struct {  
    // PublicSuffixList is the public suffix list that determines whether  
    // an HTTP server can set a cookie for a domain.  
    //  
    // A nil value is valid and may be useful for testing but it is not  
    // secure: it means that the HTTP server for foo.co.uk can set a cookie  
    // for bar.co.uk.  
    PublicSuffixList PublicSuffixList  
}
```

Options are the options for creating a new Jar.

type **PublicSuffixList**

```
type PublicSuffixList interface {  
    // PublicSuffix returns the public suffix of domain.  
    //  
    // TODO: specify which of the caller and callee is responsible for IP  
    // addresses, for leading and trailing dots, for case sensitivity, and  
    // for IDN/Punycode.  
    PublicSuffix(domain string) string  
  
    // String returns a description of the source of this public suffix  
    // list. The description will typically contain something like a time  
    // stamp or version number.  
    String() string  
}
```

PublicSuffixList provides the public suffix of a domain. For example:

- the public suffix of "example.com" is "com",
- the public suffix of "foo1.foo2.foo3.co.uk" is "co.uk", and
- the public suffix of "bar.pvt.k12.ma.us" is "pvt.k12.ma.us".

Implementations of PublicSuffixList must be safe for concurrent use by multiple goroutines.

An implementation that always returns "" is valid and may be useful for testing but it is not secure: it means that the HTTP server for foo.com can set a cookie for bar.com.

A public suffix list implementation is in the package golang.org/x/net/publicsuffix.



Source Files

[View all](#) 

[jar.go](#)

[punycode.go](#)

Why Go

[Use Cases](#)

[Case Studies](#)

Get Started

[Playground](#)

[Tour](#)

[Stack Overflow](#)

[Help](#)

Packages

[Standard Library](#)

[About Go Packages](#)

About

[Download](#)

[Blog](#)

[Issue Tracker](#)

[Release Notes](#)

[Brand Guidelines](#)

[Code of Conduct](#)

Connect

[Twitter](#)

[GitHub](#)

[Slack](#)

[r/golang](#)

[Meetup](#)

[Golang Weekly](#)

[Copyright](#)

[Terms of Service](#)

[Privacy Policy](#)

[Report an Issue](#)



Google