hash

package standard library

Version: go1.20.1 Latest | Published: Feb 14, 2023 | License: BSD-3-Clause | Imports: 1 | Imported by: 37,957

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 hash provides interfaces for hash functions.

▶ Example (BinaryMarshaler)

Index

type Hash
type Hash32
type Hash64

Examples

Package (BinaryMarshaler)

Constants

This section is empty.

Variables

This section is empty.

Functions

This section is empty.

Types

type Hash

```
type Hash interface {
    // Write (via the embedded io.Writer interface) adds more data to the running hash.
    // It never returns an error.
    io.Writer

    // Sum appends the current hash to b and returns the resulting slice.
    // It does not change the underlying hash state.
    Sum(b []byte) []byte

    // Reset resets the Hash to its initial state.
    Reset()

    // Size returns the number of bytes Sum will return.
    Size() int

    // BlockSize returns the hash's underlying block size.
    // The Write method must be able to accept any amount
    // of data, but it may operate more efficiently if all writes
    // are a multiple of the block size.
    BlockSize() int
}
```

Hash is the common interface implemented by all hash functions.

Hash implementations in the standard library (e.g. hash/crc32 and crypto/sha256) implement the encoding.BinaryMarshaler and encoding.BinaryUnmarshaler interfaces. Marshaling a hash implementation allows its internal state to be saved and used for additional processing later, without having to re-write the data previously written to the hash. The hash state may contain portions of the input in its original form, which users are expected to handle for any possible security implications.

Compatibility: Any future changes to hash or crypto packages will endeavor to maintain compatibility with state encoded using previous versions. That is, any released versions of the packages should be able to decode data written with any previously released version, subject to issues such as security fixes. See the Go compatibility document for background: https://golang.org/doc/go1compat

type Hash32

```
type Hash32 interface {
    Hash
    Sum32() uint32
}
```

Hash32 is the common interface implemented by all 32-bit hash functions.

type Hash64

```
type Hash64 interface {
    Hash
    Sum64() uint64
}
```

Hash64 is the common interface implemented by all 64-bit hash functions.

Source Files

View all

hash.go

Directories

adler32

Package adler32 implements the Adler-32 checksum.

crc32

Package crc32 implements the 32-bit cyclic redundancy check, or CRC-32, checksum.

crc64

Package crc64 implements the 64-bit cyclic redundancy check, or CRC-64, checksum.

fnv

Package fnv implements FNV-1 and FNV-1a, non-cryptographic hash functions created by Glenn Fowler, Landon Curt Noll, and Phong Vo.

maphash

Package maphash provides hash functions on byte sequences.

Twitter

GitHub

Slack

r/golang

Meetup

Golang Weekly

Report an Issue

Google