encoding

package standard library

Version: go1.20.1 Latest | Published: Feb 14, 2023 | License: BSD-3-Clause | Imports: 0 | Imported by: 8,988

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 encoding defines interfaces shared by other packages that convert data to and from byte-level and textual representations. Packages that check for these interfaces include encoding/gob, encoding/json, and encoding/xml. As a result, implementing an interface once can make a type useful in multiple encodings. Standard types that implement these interfaces include time.Time and net.IP. The interfaces come in pairs that produce and consume encoded data.

Index

Constants

This section is empty.

Variables

This section is empty.

Functions

This section is empty.

Types

type BinaryMarshaler

```
type BinaryMarshaler interface {
    MarshalBinary() (data []byte, err error)
}
```

BinaryMarshaler is the interface implemented by an object that can marshal itself into a binary form.

MarshalBinary encodes the receiver into a binary form and returns the result.

type BinaryUnmarshaler

```
type BinaryUnmarshaler interface {
    UnmarshalBinary(data []byte) error
}
```

BinaryUnmarshaler is the interface implemented by an object that can unmarshal a binary representation of itself.

UnmarshalBinary must be able to decode the form generated by MarshalBinary. UnmarshalBinary must copy the data if it wishes to retain the data after returning.

type TextMarshaler

```
type TextMarshaler interface {
    MarshalText() (text []byte, err error)
}
```

TextMarshaler is the interface implemented by an object that can marshal itself into a textual form.

MarshalText encodes the receiver into UTF-8-encoded text and returns the result.

type TextUnmarshaler

```
type TextUnmarshaler interface {
    UnmarshalText(text []byte) error
}
```

TextUnmarshaler is the interface implemented by an object that can unmarshal a textual representation of itself.

UnmarshalText must be able to decode the form generated by MarshalText. UnmarshalText must copy the text if it wishes to retain the text after returning.

Source Files

View all

encoding.go

Directories

ascii85

Package ascii85 implements the ascii85 data encoding as used in the btoa tool and Adobe's PostScript and PDF document formats.

asn1

Package asn1 implements parsing of DER-encoded ASN.1 data structures, as defined in ITU-T Rec X.690.

base32

Package base32 implements base32 encoding as specified by RFC 4648.

base64

Package base64 implements base64 encoding as specified by RFC 4648.

binary

Package binary implements simple translation between numbers and byte sequences and encoding and decoding of varints.

csv

Package csv reads and writes comma-separated values (CSV) files.

gob

Package gob manages streams of gobs - binary values exchanged between an Encoder (transmitter) and a Decoder (receiver).

hex

Package hex implements hexadecimal encoding and decoding.

json

Package json implements encoding and decoding of JSON as defined in RFC 7159.

pem

Package pem implements the PEM data encoding, which originated in Privacy Enhanced Mail.

xml

Package xml implements a simple XML 1.0 parser that understands XML name spaces.

Google