Discover Packages > Standard library > encoding > pem

pem

package standard library

Version: go1.20.1 Latest | Published: Feb 14, 2023 | License: BSD-3-Clause | Imports: 6 | Imported by: 33,814

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 pem implements the PEM data encoding, which originated in Privacy Enhanced Mail. The most common use of PEM encoding today is in TLS keys and certificates. See RFC 1421.

Index

Examples

Constants

This section is empty.

Variables

This section is empty.

Functions

func Encode

```
func Encode(out io.Writer, b *Block) error
```

Encode writes the PEM encoding of b to out.

▶ Example

func EncodeToMemory

```
func EncodeToMemory(b *Block) []byte
```

EncodeToMemory returns the PEM encoding of b.

If b has invalid headers and cannot be encoded, EncodeToMemory returns nil. If it is important to report details about this error case, use Encode instead.

Types

type Block

```
type Block struct {
    Type    string            // The type, taken from the preamble (i.e. "RSA PRIVATE KE
    Headers map[string]string // Optional headers.
    Bytes   []byte            // The decoded bytes of the contents. Typically a DER enc
}
```

A Block represents a PEM encoded structure.

The encoded form is:

```
-----BEGIN Type-----
Headers
base64-encoded Bytes
-----END Type-----
```

where Headers is a possibly empty sequence of Key: Value lines.

func Decode

```
func Decode(data []byte) (p *Block, rest []byte)
```

Decode will find the next PEM formatted block (certificate, private key etc) in the input. It returns that block and the remainder of the input. If no PEM data is found, p is nil and the whole of the input is returned in rest.

▶ Example

Source Files

View all

pem.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