rand   package   standard library

Version: go1.20.1 Latest | Published: Feb 14, 2023 | License: BSD-3-Clause | Imports: 12 |
Imported by: 102,467

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**   Rendered for   linux/amd64 ▼

Overview

Package rand implements a cryptographically secure random number generator.

Index

Variables
func Int(rand io.Reader, max *big.Int) (n *big.Int, err error)
func Prime(rand io.Reader, bits int) (*big.Int, error)
func Read(b []byte) (n int, err error)

Examples

Read

Constants

This section is empty.

Variables

View Source

```
var Reader io.Reader
```

Reader is a global, shared instance of a cryptographically secure random number generator.

On Linux, FreeBSD, Dragonfly and Solaris, Reader uses getrandom(2) if available, /dev/urandom otherwise. On OpenBSD and macOS, Reader uses getentropy(2). On other Unix-like systems, Reader reads from /dev/urandom. On Windows systems, Reader uses the RtlGenRandom API. On Wasm, Reader uses the Web Crypto API.

Functions

func Int

```
func Int(rand io.Reader, max *big.Int) (n *big.Int, err error)
```

Int returns a uniform random value in [0, max). It panics if max <= 0.

func Prime

```
func Prime(rand io.Reader, bits int) (*big.Int, error)
```

Prime returns a number of the given bit length that is prime with high probability. Prime will return error for any error returned by rand.Read or if bits < 2.

func Read

```
func Read(b []byte) (n int, err error)
```

Read is a helper function that calls Reader.Read using io.ReadFull. On return, n == len(b) if and only if err == nil.

▶ Example

Types

This section is empty.

📄 Source Files

View all ⬈

rand.go

rand_unix.go

rand_getrandom.go

util.go

Why Go

Get Started

Packages

About

Use Cases

Playground

Standard Library

Download

Case Studies

Tour

About Go Packages

Blog

Stack Overflow

Issue Tracker

Help

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