

UUID



Package `uuid` provides a pure Go implementation of Universally Unique Identifiers (UUID) variant as defined in RFC-4122. This package supports both the creation and parsing of UUIDs in different formats.

This package supports the following UUID versions:

- Version 1, based on timestamp and MAC address (RFC-4122)
- Version 3, based on MD5 hashing of a named value (RFC-4122)
- Version 4, based on random numbers (RFC-4122)
- Version 5, based on SHA-1 hashing of a named value (RFC-4122)

Project History

This project was originally forked from the github.com/satori/go.uuid repository after it appeared to be no longer maintained, while exhibiting [critical flaws](#). We have decided to take over this project to ensure it receives regular maintenance for the benefit of the larger Go community.

We'd like to thank Maxim Bubliss for his hard work on the original iteration of the package.

License

This source code of this package is released under the MIT License. Please see the [LICENSE](#) for the full content of the license.

Recommended Package Version

We recommend using v2.0.0+ of this package, as versions prior to 2.0.0 were created before our fork of the original package and have some known deficiencies.

Installation

It is recommended to use a package manager like `dep` that understands tagged releases of a package, as well as semantic versioning.

If you are unable to make use of a dependency manager with your project, you can use the `go get` command to download it directly:

```
$ go get github.com/gofrs/uuid
```

Requirements

Due to subtests not being supported in older versions of Go, this package is only regularly tested against Go 1.7+. This package may work perfectly fine with Go 1.2+, but support for these older versions is not actively maintained.

Go 1.11 Modules

As of v3.2.0, this repository no longer adopts Go modules, and v3.2.0 no longer has a `go.mod` file. As a result, v3.2.0 also drops support for the `github.com/gofrs/uuid/v3` import path. Only module-based consumers are impacted. With the v3.2.0 release, *all* gofrs/uuid consumers should use the `github.com/gofrs/uuid` import path.

An existing module-based consumer will continue to be able to build using the `github.com/gofrs/uuid/v3` import path using any valid consumer `go.mod` that worked prior to the publishing of v3.2.0, but any module-based consumer should start using the `github.com/gofrs/uuid` import path when possible and *must* use the `github.com/gofrs/uuid` import path prior to upgrading to v3.2.0.

Please refer to [Issue #61](#) and [Issue #66](#) for more details.

Usage

Here is a quick overview of how to use this package. For more detailed documentation, please see the [GoDoc Page](#).

```
package main

import (
    "log"

    "github.com/gofrs/uuid"
)

// Create a Version 4 UUID, panicking on error.
// Use this form to initialize package-level variables.
var u1 = uuid.Must(uuid.NewV4())

func main() {
    // Create a Version 4 UUID.
    u2, err := uuid.NewV4()
    if err != nil {
        log.Fatalf("failed to generate UUID: %v", err)
    }
    log.Printf("generated Version 4 UUID %v", u2)

    // Parse a UUID from a string.
    s := "6ba7b810-9dad-11d1-80b4-00c04fd430c8"
    u3, err := uuid.FromString(s)
    if err != nil {
        log.Fatalf("failed to parse UUID %q: %v", s, err)
    }
    log.Printf("successfully parsed UUID %v", u3)
}
```

References

- [RFC-4122](#)
- [DCE 1.1: Authentication and Security Services](#)