go-digest

GO reference go report A+ build passing

Common digest package used across the container ecosystem.

Please see the godoc for more information.

What is a digest?

A digest is just a hash.

The most common use case for a digest is to create a content identifier for use in Content Addressable Storage systems:

```
id := digest.FromBytes([]byte("my content"))
```

In the example above, the id can be used to uniquely identify the byte slice "my content". This allows two disparate applications to agree on a verifiable identifier without having to trust one another.

An identifying digest can be verified, as follows:

```
if id != digest.FromBytes([]byte("my content")) {
  return errors.New("the content has changed!")
}
```

A `Verifier` type can be used to handle cases where an `io.Reader` makes more sense:

```
rd := getContent()
verifier := id.Verifier()
io.Copy(verifier, rd)

if !verifier.Verified() {
  return errors.New("the content has changed!")
}
```

Using Merkle DAGs, this can power a rich, safe, content distribution system.

Usage

While the godoc is considered the best resource, a few important items need to be called out when using this package.

1. Make sure to import the hash implementations into your application or the package will panic. You should have something like the following in the main (or other entrypoint) of your application:

```
   import (
   	_ "crypto/sha256"
   	_ "crypto/sha512"
   )
   ```

   This may seem inconvenient but it allows you replace the hash implementations with others, such as https://github.com/stevvooe/resumable.

2. Even though `digest.Digest` may be assemblable as a string, *always* verify your input with `digest.Parse` or use `Digest.Validate` when accepting untrusted input. While there are measures to avoid common problems, this will ensure you have valid digests in the rest of your application.

3. While alternative encodings of hash values (digests) are possible (for example, base64), this package deals exclusively with hex-encoded digests.

Stability

The Go API, at this stage, is considered stable, unless otherwise noted.

As always, before using a package export, read the godoc.

Contributing

This package is considered fairly complete. It has been in production in thousands (millions?) of deployments and is fairly battle-hardened. New additions will be met with skepticism. If you think there is a missing feature, please file a bug clearly describing the problem and the alternatives you tried before submitting a PR.

Code of Conduct

Participation in the OpenContainers community is governed by OpenContainer's Code of Conduct.

Security

If you find an issue, please follow the security protocol to report it.

Copyright and license