GO reference build passing codecov 100% go report A+

atomic

Simple wrappers for primitive types to enforce atomic access.

Installation

```
$ go get -u go.uber.org/atomic@v1
```

Legacy Import Path

As of v1.5.0, the import path `go.uber.org/atomic` is the only supported way of using this package. If you are using Go modules, this package will fail to compile with the legacy import path path `github.com/uber-go/atomic`.

We recommend migrating your code to the new import path but if you're unable to do so, or if your dependencies are still using the old import path, you will have to add a `replace` directive to your `go.mod` file downgrading the legacy import path to an older version.

```
replace github.com/uber-go/atomic => github.com/uber-go/atomic v1.4.0
```

You can do so automatically by running the following command.

```
$ go mod edit -replace github.com/uber-go/atomic=github.com/uber-go/atomic@v1.4.0
```

Usage

The standard library's `sync/atomic` is powerful, but it's easy to forget which variables must be accessed atomically. `go.uber.org/atomic` preserves all the functionality of the standard library, but wraps the primitive types to provide a safer, more convenient API.

```
var atom atomic.Uint32
atom.Store(42)
atom.Sub(2)
atom.CAS(40, 11)
```

See the documentation for a complete API specification.

Development Status

Stable.
