OpenTelemetry-Go

ci passing GO reference go report A+ slack @cncf/otel-go

The Go OpenTelemetry implementation.

Project Status

Warning: this project is currently in a pre-GA phase. Backwards incompatible changes may be introduced in subsequent minor version releases as we work to track the evolving OpenTelemetry specification and user feedback.

Our progress towards a GA release candidate is tracked in this project board. This release candidate will follow semantic versioning and will be released with a major version greater than zero.

Progress and status specific to this repository is tracked in our local project boards and milestones.

Project versioning information and stability guarantees can be found in the versioning documentation.

Compatibility

This project is tested on the following systems.

OS	Go Version	Architecture
Ubuntu	1.15	amd64
Ubuntu	1.14	amd64
Ubuntu	1.15	386
Ubuntu	1.14	386
MacOS	1.15	amd64
MacOS	1.14	amd64
Windows	1.15	amd64
Windows	1.14	amd64
Windows	1.15	386
Windows	1.14	386

While this project should work for other systems, no compatibility guarantees are made for those systems currently.

Getting Started

You can find a getting started guide on opentelemetry.io.

OpenTelemetry's goal is to provide a single set of APIs to capture distributed traces and metrics from your application and send them to an observability platform. This project allows you to do just that for applications written in Go. There are two steps to this process: instrument your application, and configure an exporter.

Instrumentation

To start capturing distributed traces and metric events from your application it first needs to be instrumented. The easiest way to do this is by using an instrumentation library for your code. Be sure to check out the officially supported instrumentation libraries.

If you need to extend the telemetry an instrumentation library provides or want to build your own instrumentation for your application directly you will need to use the go.opentelemetry.io/otel/api package. The included examples are a good way to see some practical uses of this process.

Export

Now that your application is instrumented to collect telemetry, it needs an export pipeline to send that telemetry to an observability platform.

You can find officially supported exporters here and in the companion contrib repository. Additionally, there are many vendor specific or 3rd party exporters for OpenTelemetry. These exporters are broken down by trace and metric support.

Contributing

See the contributing documentation.