Client Libraries

Official libraries

Language	Library
go	jaeger-client-go
java	jaeger-client-java
node	jaeger-client-node
python	jaeger-client-python
C++	cpp-client

For an introduction into instrumenting your service with the OpenTracing API using the Jaeger clients, see OpenTracing Tutorials

For a deep dive into how to instrument a Go service, look at Tracing HTTP request latency.

Initializing Jaeger Tracer

The initialization syntax is slightly different in each languages, please refer to the README's in the respective repositories. The general pattern is to not create the Tracer explicitly, but use a Configuration class to do that. Configuration allows simpler parameterization of the Tracer, such as changing the default sampler or the location of Jaeger agent.

EMSGSIZE and UDP buffer limits

By default Jaeger libraries use a UDP sender to report finished spans to `jaeger-agent` sidecar. The default max packet size is 65,000 bytes, which can be transmitted without segmentation when connecting to the agent via loopback interface. However, some OSs (in particular, MacOS), limit the max buffer size for UDP packets, as raised in [this GitHub issue](https://github.com/uber/jaeger-client-node/issues/124). If you run into issue with `EMSGSIZE` errors, consider raising the limits in your kernel (see the issue for examples). You can also configure the client libraries to use a smaller max packet size, but that may cause issues if you have large spans, e.g. if you log big chunks of data. Spans that exceed max packet size are dropped by the clients (with metrics emitted to indicate that). Another alternative is to use non-UDP transports, such as HttpSender in Java (not currently available for all languages).

OpenTracing Contributions

See the OpenTracing contributions repository on Github for more libraries.

Propagation Format

When `SpanContext` is encoded on the wire, Jaeger client libraries default to the encoding specified below.

Trace/Span Identity

Key

`uber-trace-id`

- Case-insensitive in HTTP
- Lower-case in protocols that preserve header case

Value

`{trace-id}:{span-id}:{parent-span-id}:{flags}`

- `{trace-id}`
- 64-bit or 128-bit random number in base16 format
- Can be variable length, shorter values are 0-padded on the left
- Clients in some languages support 128-bit, migration pending
- Value of 0 is invalid
- `{span-id}`
- 64-bit random number in base16 format
- `{parent-span-id}`
- 64-bit value in base16 format representing parent span id
- Deprecated, most Jaeger clients ignore on the receiving side, but still include it on the sending side
- 0 value is valid and means “root span” (when not ignored)
- `{flags}`
- One byte bitmap, as two hex digits
- Bit 1 (right-most, least significant) is “sampled” flag
- 1 means the trace is sampled and all downstream services are advised to respect that
- 0 means the trace is not sampled and all downstream services are advised to respect that
- We’re considering a new feature that allows downstream services to upsample if they find their tracing level is too low
- Bit 2 is “debug” flag
- Debug flag implies sampled flag
- Instructs the backend to try really hard not to drop this trace
- Other bits are unused

Baggage

- Key: `uberctx-{baggage-key}`
- Value: url-encoded string

- Limitation: since HTTP headers don't preserve the case, Jaeger recommends baggage keys to be lowercase-snake-case, e.g. `my-baggage-key-1`.