

Please visit website: <http://cxyroad.com>

golang tracing

=====

Tracing

tracing

1. ****
2. ****
3. ****
4. ****

OpenTelemetry

OpenTelemetry tracing tracing tracing

tracing Trace “” Trace Span Trace Span [Google Dapper](http://cxyroad.com/"https://research.google.com/pubs/pub36356.html")

* Trace Span Span

Span “” SpanID ParentSpanID

* SpanID ParentSpanID SpanID
* Span TraceIDSpanID Trace PropagationTrace
Propagation Trace
* ServiceMesh Trace Propagation
* OpenTracing Google OpenCensus
* OpenTracing OpenCensus
OpenTelemetryOpenTelemetry Trace

OpenTelemetry Trace SDKTraces/Metrics/Logs
[OpenTelemetry](http://cxyroad.com/"https://developer.aliyun.com/article/766070")

SLS

sls OpenTelemetry tracing

sls

```
* OpenTelemetryJaeger httpsgrpc ZipkinOpenCensus
Trace
* OpenTelemetry Collector OpenTelemetryJaegerZipkin
OpenCensusAWS X-RaySignalFXSplunk Trace
* Logtail SkyWalking Trace
* Trace Trace OpenTelemetry
```

push tracing OpenTelemetry sdk

```
### ****
```

```
1. ****
```

```
...
```

```
[data.tracing]
  traceExporterEndpoint = ""
  metricExporterEndpoint = ""
  slsProject = ""
  slsInstance = ""
  accessKeyId = ""
  accessKeySecret = ""
```

```
...
```

sls

```
**provider**
```

```
...
```

```
var (
```

```

MetricPushInterval    = 20
Namespace             string = "xxx"
// MetricName is the name of the compiled software.
MetricName = "xxx"
// Name is the name of the compiled software.
Name = "xxx"
// Version is the version of the compiled software.
Version string = "v0.1.0"
)

```

```

func initTracerConfig(tracingConfig *conf.Tracing)
*provider.Config {
    // Namespace, Name, Version
    slsConfig, err :=
provider.NewConfig(provider.WithServiceName(Name),
    provider.WithServiceVersion(Version),
    provider.WithServiceNamespace(Namespace),
provider.WithTraceExporterEndpoint(tracingConfig.TraceE
xporterEndpoint),
provider.WithMetricExporterEndpoint(tracingConfig.Metric
ExporterEndpoint),
    provider.WithSLSConfig(tracingConfig.SlsProject,
tracingConfig.SlsInstance, tracingConfig.AccessKeyId,
tracingConfig.AccessKeySecret))
    // panic
    if err != nil {
        panic(err)
    }
    return slsConfig
}

```

...

1. ****main.go****

...

```

func main() {
    flag.Parse()
    c := config.New(config.WithPath(confPath))
    if err := c.Load(); err != nil {
        panic(err)
    }
    if err := c.Scan(&conf.Cfg); err != nil {
        panic(err)
    }
}

```

```

//
// tracing
slsConfig := initTracerConfig(&conf.Cfg.Data.Tracing)
if err := provider.Start(slsConfig); err != nil {
    panic(err)
}
defer provider.Shutdown(slsConfig)
// ...
}
...

### **gin tracing**

gin

tracing

...

go get
"go.opentelemetry.io/contrib/instrumentation/github.com/gi
n-gonic/gin/otelgin"
...

gin

...

func main() {
    // .....
    r := gin.Default()
    r.Use(otelgin.Middleware(serviceName))
    // .....
}
...

### **Redis tracing**

go-redis OpenTelemetry

```

redisotel

...

```
go get "github.com/go-redis/redis/extra/redisotel/v8"
```

...

redis-cli hook

...

```
func NewRedisClient(conf *conf.Data) *redis.Client {  
    // ...  
    // tracing  
    client.AddHook(redisotel.NewTracingHook())  
    // ...  
    return client  
}
```

...

MySQL tracing

gorm OpenTelemetry

gromotel

...

```
go get "gorm.io/plugin/opentelemetry/tracing"
```

...

db-cli

...

```
func NewDB(conf *conf.Data, logger log.Logger,  
zapLogger *zap.Logger) *gorm.DB {  
    // ...  
}
```

```
    if err =
db.Use(tracing.NewPlugin(tracing.WithoutMetrics())); err !=
nil {
    panic(err)
}
return db
}
```

...

push-service ** **

**trace **

sls trace sls tracingmetriclog

**** lib****

[github.com/go-gorm/ope...](http://cxyroad.com/
"https://github.com/go-gorm/opentelemetry")

[upstash.com/blog/go-red...](http://cxyroad.com/
"https://upstash.com/blog/go-redis-opentelemetry")

[github.com/redis/go-re...](http://cxyroad.com/
"https://github.com/redis/go-redis/tree/master/extra/redisotel")

[help.aliyun.com/zh/sls/user...](http://cxyroad.com/
"https://help.aliyun.com/zh/sls/user-guide/import-trace-data-from-golang-applications-to-log-service-by-using-opentelemetry-sdk-for-golang?spm=a2c4g.11186623.0.i29")

[flow.visionhope.cn/posts/opent...](http://cxyroad.com/
"https://flow.visionhope.cn/posts/opentelemetry-guide/")

**** Tracing ****

[xie.infoq.cn/article/8f4...](http://cxyroad.com/
"https://xie.infoq.cn/article/8f4b171b1992c6d95e4426230")

[developer.aliyun.com/article/785...](http://cxyroad.com/
"https://developer.aliyun.com/article/785854")

[developer.aliyun.com/article/783...](http://cxyroad.com/
"https://developer.aliyun.com/article/783270?spm=a2c6h.12873639.article-detail.10.3e2c5d17TdedkW")

: https://juejin.cn/post/7352075662453309478

