

# rpc trace中间件开发

联系QQ: 2816010068, 加入会员群

# 目录

- Opentrace 配置和初始化
- Opentrace 中间件开发

# Opentrace 配置和初始化

- 上报中心的地址
- 采样的类型
- 采样的频率

```
7  type RpcOptions struct {  
8      ConnTimeout  time.Duration  
9      WriteTimeout time.Duration  
10     ReadTimeout  time.Duration  
11     ServiceName  string  
12     //注册中心名字  
13     RegisterName string  
14     //注册中心地址  
15     RegisterAddr string  
16     //注册中心路径  
17     RegisterPath string  
18     //限流的qps  
19     MaxLimitQps int  
20     //trace report address  
21     TraceReportAddr string  
22     //trace sample type  
23     TraceSampleType string  
24     //trace sample rate  
25     TraceSampleRate float64  
26     //clientServiceName  
27     ClientServiceName string  
28 }
```

# Opentrace 配置和初始化

- 选项函数实现

```
func WithTraceReportAddr(addr string) RpcOptionFunc {  
    return func(opts *RpcOptions) {  
        opts.TraceReportAddr = addr  
    }  
}  
  
func WithTraceSampleType(stype string) RpcOptionFunc {  
    return func(opts *RpcOptions) {  
        opts.TraceSampleType = stype  
    }  
}  
  
func WithTraceSampleRate(rate float64) RpcOptionFunc {  
    return func(opts *RpcOptions) {  
        opts.TraceSampleRate = rate  
    }  
}  
  
func WithClientServiceName(name string) RpcOptionFunc {  
    return func(opts *RpcOptions) {  
        opts.ClientServiceName = name  
    }  
}
```

# Opentrace 配置和初始化

- 默认值

```
client := &KoalaClient{
    opts: &RpcOptions{
        ConnTimeout:    DefaultConnTimeout,
        WriteTimeout:    DefaultWriteTimeout,
        ReadTimeout:     DefaultReadTimeout,
        ServiceName:     serviceName,
        RegisterName:    "etcd",
        RegisterAddr:    "127.0.0.1:2379",
        RegisterPath:    "/ibinarytree/koala/service/",
        TraceReportAddr: "http://60.205.218.189:9411/api/v1/spans",
        TraceSampleType: "const",
        TraceSampleRate: 1,
        ClientServiceName: "default",
    },
    balance: loadbalance.NewRandomBalance(),
}
```

# Opentrace 中间件开发

- 关键代码：
  - 启动跟踪: `clientSpan := tracer.StartSpan(rpcMeta.ServiceName, opts...)`
  - **【跟踪】** 传播:  
`tracer.Inject(clientSpan.Context(), opentracing.HTTPHeaders, metadataTextMap(md))`
  - 结束跟踪: `clientSpan.Finish()`

# 集中测试