

<https://github.com/gogf/gf/tree/master/example/trace/inprocess>

Root Span

root spanspanspanspan

 root span/

tracerroot span

```
func main() {
    const (
        serviceName = "otlp-http-client"
        endpoint     = "tracing-analysis-dc-hz.aliyuncs.com"
        path        = "adapt_*****_*****/api/otlp/traces"
    )

    var ctx = gctx.New()
    shutdown, err := otlphttp.Init(serviceName, endpoint, path)
    if err != nil {
        g.Log().Fatal(ctx, err)
    }
    defer shutdown()

    ctx, span := gtrace.NewSpan(ctx, "main")
    defer span.End()

    // Trace 1.
    user1 := GetUser(ctx, 1)
    g.Dump(user1)

    // Trace 2.
    user100 := GetUser(ctx, 100)
    g.Dump(user100)
}
```

root spanspancontextGetUserGetUser

Span

Content Menu

-
- [Root Span](#)
- [Span](#)
-

```

// GetUser retrieves and returns hard coded user data for demonstration.
func GetUser(ctx context.Context, id int) g.Map {
    ctx, span := gtrace.NewSpan(ctx, "GetUser")
    defer span.End()
    m := g.Map{}
    gutil.MapMerge(
        m,
        GetInfo(ctx, id),
        GetDetail(ctx, id),
        GetScores(ctx, id),
    )
    return m
}

// GetInfo retrieves and returns hard coded user info for demonstration.
func GetInfo(ctx context.Context, id int) g.Map {
    ctx, span := gtrace.NewSpan(ctx, "GetInfo")
    defer span.End()
    if id == 100 {
        return g.Map{
            "id":      100,
            "name":     "john",
            "gender": 1,
        }
    }
    return nil
}

// GetDetail retrieves and returns hard coded user detail for demonstration.
func GetDetail(ctx context.Context, id int) g.Map {
    ctx, span := gtrace.NewSpan(ctx, "GetDetail")
    defer span.End()
    if id == 100 {
        return g.Map{
            "site": "https://goframe.org",
            "email": "john@goframe.org",
        }
    }
    return nil
}

// GetScores retrieves and returns hard coded user scores for demonstration.
func GetScores(ctx context.Context, id int) g.Map {
    ctx, span := gtrace.NewSpan(ctx, "GetScores")
    defer span.End()
    if id == 100 {
        return g.Map{
            "math":      100,
            "english": 60,
            "chinese": 50,
        }
    }
    return nil
}

```

ctx/Span

```

ctx, span := gtrace.NewSpan(ctx, "xxx")
defer span.End()

```

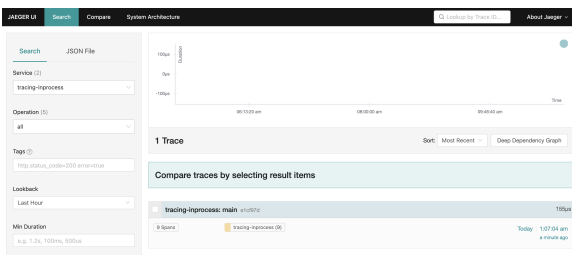
defer span.End() span gtrace.NewSpan spanName

```
→ gf-tracing git:(master) go run inprocess/main.go
{}

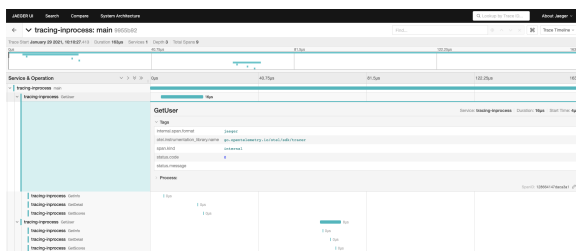
{
    "chinese": 50,
    "email": "john@goframe.org",
    "english": 60,
    "gender": 1,
    "id": 100,
    "math": 100,
    "name": "john",
    "site": "https://goframe.org"
}

→ gf-tracing git:(master) █
```

Jaeger UI: <http://localhost:16686/search>



spanAttributesEvents



tracing-inprocess tracer tracer main root spanspan root spanspan GetUser GetUser GetUser
erGetIndoGetDetailGetScores

spanTagsProcessOpenTelemetryAttributesEvents