-HTTP-

HTTP+DB+Redis+Logging

https://github.com/gogf/gf/tree/master/example/trace/http_with_db


```
package main

import (
        "github.com/gogf/gf/contrib/trace/otlphttp/v2"
        "github.com/gogf/gf/v2/database/gdb"
        "github.com/gogf/gf/v2/frame/g"
        "github.com/gogf/gf/v2/net/ghttp"
        "github.com/gogf/gf/v2/net/gtrace"
        "github.com/gogf/gf/v2/os/gctx"
)

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

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

    StartRequests()
}

func StartRequests() {
        ctx, span := gtrace.NewSpan(gctx.New(), "StartRequests")
        defer span.End()

        var (
                err    error
                client = g.Client()
        )
        // Add user info.
        var insertRes = struct {
                ghttp.DefaultHandlerResponse
                Data struct{ Id int64 } `json:"data"`
        }{}
        err = client.PostVar(ctx, "http://127.0.0.1:8199/user/insert", g.
Map{
                "name": "john",
        }).Scan(&insertRes)
        if err != nil {
                panic(err)
        }
        g.Log().Info(ctx, "insert result:", insertRes)
        if insertRes.Data.Id == 0 {
                g.Log().Error(ctx, "retrieve empty id string")
                return
        }

        // Query user info.
        var queryRes = struct {
                ghttp.DefaultHandlerResponse
                Data struct{ User gdb.Record } `json:"data"`
        }{}
        err = client.GetVar(ctx, "http://127.0.0.1:8199/user/query", g.Map{
```

```
                "id": insertRes.Data.Id,
        }).Scan(&queryRes)
        if err != nil {
                panic(err)
        }
        g.Log().Info(ctx, "query result:", queryRes)

        // Delete user info.
        var deleteRes = struct {
                ghttp.DefaultHandlerResponse
        }{}
        err = client.PostVar(ctx, "http://127.0.0.1:8199/user/delete", g.
Map{
                "id": insertRes.Data.Id,
        }).Scan(&deleteRes)
        if err != nil {
                panic(err)
        }
        g.Log().Info(ctx, "delete result:", deleteRes)
}
```

1. jaeger.InitJaeger
2. HTTP3
    a. /user/insert ID
    b. /user/query ID
    c. /user/delete ID

```
package main

import (
        "context"
        "fmt"
        "time"

        "github.com/gogf/gf/contrib/trace/otlphttp/v2"
        "github.com/gogf/gf/v2/database/gdb"
        "github.com/gogf/gf/v2/frame/g"
        "github.com/gogf/gf/v2/net/ghttp"
        "github.com/gogf/gf/v2/os/gcache"
        "github.com/gogf/gf/v2/os/gctx"
)

type cTrace struct{}

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

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

    // Set ORM cache adapter with redis.
        g.DB().GetCache().SetAdapter(gcache.NewAdapterRedis(g.Redis()))

        // Start HTTP server.
        s := g.Server()
        s.Use(ghttp.MiddlewareHandlerResponse)
        s.Group("/", func(group *ghttp.RouterGroup) {
```

```
                group.ALL("/user", new(cTrace))
        })
        s.SetPort(8199)
        s.Run()
}

type InsertReq struct {
        Name string `v:"required#Please input user name."`
}
type InsertRes struct {
        Id int64
}

// Insert is a route handler for inserting user info into database.
func (c *cTrace) Insert(ctx context.Context, req *InsertReq) (res
*InsertRes, err error) {
        result, err := g.Model("user").Ctx(ctx).Insert(req)
        if err != nil {
                return nil, err
        }
        id, _ := result.LastInsertId()
        res = &InsertRes{
                Id: id,
        }
        return
}

type QueryReq struct {
        Id int `v:"min:1#User id is required for querying"`
}
type QueryRes struct {
        User gdb.Record
}

// Query is a route handler for querying user info. It firstly retrieves
the info from redis,
// if there's nothing in the redis, it then does db select.
func (c *cTrace) Query(ctx context.Context, req *QueryReq) (res *QueryRes,
err error) {
        one, err := g.Model("user").Ctx(ctx).Cache(gdb.CacheOption{
                Duration: 5 * time.Second,
                Name:     c.userCacheKey(req.Id),
                Force:    false,
        }).WherePri(req.Id).One()
        if err != nil {
                return nil, err
        }
        res = &QueryRes{
                User: one,
        }
        return
}

type DeleteReq struct {
        Id int `v:"min:1#User id is required for deleting."`
}
type DeleteRes struct{}

// Delete is a route handler for deleting specified user info.
func (c *cTrace) Delete(ctx context.Context, req *DeleteReq) (res
*DeleteRes, err error) {
        _, err = g.Model("user").Ctx(ctx).Cache(gdb.CacheOption{
                Duration: -1,
                Name:     c.userCacheKey(req.Id),
                Force:    false,
        }).WherePri(req.Id).Delete()
        if err != nil {
                return nil, err
        }
        return
}
```

```
func (c *cTrace) userCacheKey(id int) string {
        return fmt.Sprintf(`userInfo:%d`, id)
}
```

1. jaeger.InitJaeger
2. ORMRedis
3. redis -

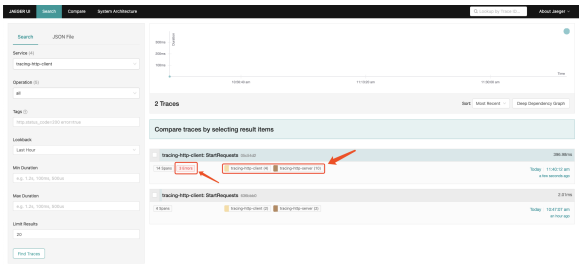
```
g.DB().GetCache().SetAdapter(gcache.NewAdapterRedis(g.Redis()))
```

4. ORMCtxormTracing
5. ORMCacheredisCacheredisORM ORM-

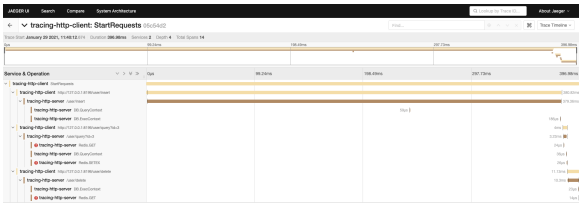
```
→  gf-tracing git:(master) go run http+db+redis+log/server/main.go
2021-01-29 11:40:08.163 70836: http server started listening on [:8199]

  SERVER  | DOMAIN  | ADDRESS | METHOD |    ROUTE     |          HANDLER           |          MIDDLEWARE
|---------|---------|---------|--------|--------------|----------------------------|-------------------------------|
  default | default | :8199   | ALL    | /user/delete | main.(*tracingApi).Delete  | ghttp.MiddlewareServerTracing
|---------|---------|---------|--------|--------------|----------------------------|-------------------------------|
  default | default | :8199   | ALL    | /user/insert | main.(*tracingApi).Insert  | ghttp.MiddlewareServerTracing
|---------|---------|---------|--------|--------------|----------------------------|-------------------------------|
  default | default | :8199   | ALL    | /user/query  | main.(*tracingApi).Query   | ghttp.MiddlewareServerTracing
|---------|---------|---------|--------|--------------|----------------------------|-------------------------------|

2021-01-29 11:40:12.915 [DEBU] {TraceID:05c54d2737e4bc8ff1c7d03b87c25b10} [240 ms] [default] SHOW FULL COLUMNS FROM `user`
2021-01-29 11:40:13.054 [DEBU] {TraceID:05c54d2737e4bc8ff1c7d03b87c25b10} [138 ms] [default] INSERT INTO `user`(`name`) VALUES('john')
2021-01-29 11:40:13.059 [DEBU] {TraceID:05c54d2737e4bc8ff1c7d03b87c25b10} [  3 ms] [default] SELECT * FROM `user` WHERE `uid`=3 LIMIT 1
2021-01-29 11:40:13.070 [DEBU] {TraceID:05c54d2737e4bc8ff1c7d03b87c25b10} [ 10 ms] [default] DELETE FROM `user` WHERE `uid`=3
```

```
→  gf-tracing git:(master) go run http+db+redis+log/client/main.go
2021-01-29 11:40:13.055 {TraceID:05c54d2737e4bc8ff1c7d03b87c25b10} insert: 3
2021-01-29 11:40:13.059 {TraceID:05c54d2737e4bc8ff1c7d03b87c25b10} query: 3 {"name":"john","uid":3}
2021-01-29 11:40:13.071 {TraceID:05c54d2737e4bc8ff1c7d03b87c25b10} delete: 3 ok
→  gf-tracing git:(master)
```

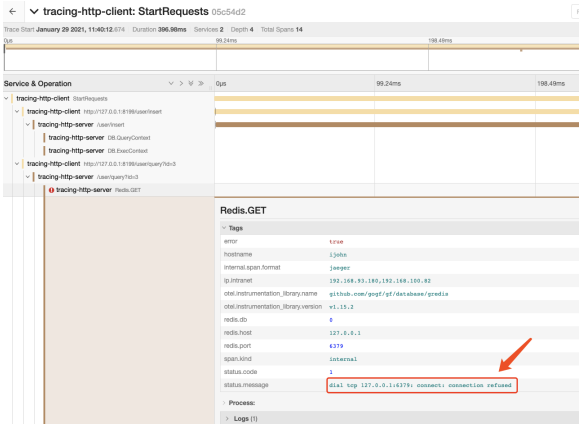
Jaeger



14span4span10spanspan3errors



redisredisspan

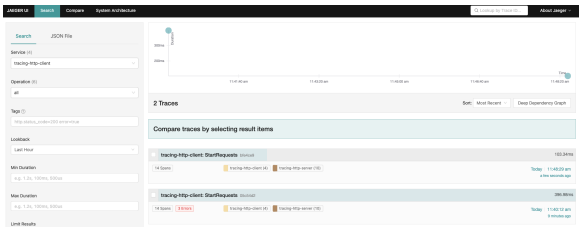


redisormredis serverredis server

```
➜  ~ redis-server
70947:C 29 Jan 2021 11:47:35.871 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
70947:C 29 Jan 2021 11:47:35.872 # Redis version=5.0.3, bits=64, commit=00000000, m
70947:C 29 Jan 2021 11:47:35.872 # Warning: no config file specified, using the def
edis.conf
70947:M 29 Jan 2021 11:47:35.873 * Increased maximum number of open files to 10032
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 5.0.3 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in standalone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 70947
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

70947:M 29 Jan 2021 11:47:35.875 # Server initialized
70947:M 29 Jan 2021 11:47:35.876 * DB loaded from disk: 0.001 seconds
70947:M 29 Jan 2021 11:47:35.876 * Ready to accept connections
```

jaeger

```
➜  gf-tracing git:(master) go run http+db+redis+log/client/main.go
2021-01-29 11:48:29.812 {TraceID:bfe4ce92c51875b2b68f275492427308} insert: 4
2021-01-29 11:48:29.817 {TraceID:bfe4ce92c51875b2b68f275492427308} query: 4 {"name":"john","uid":4}
2021-01-29 11:48:29.832 {TraceID:bfe4ce92c51875b2b68f275492427308} delete: 4 ok
➜  gf-tracing git:(master)
```

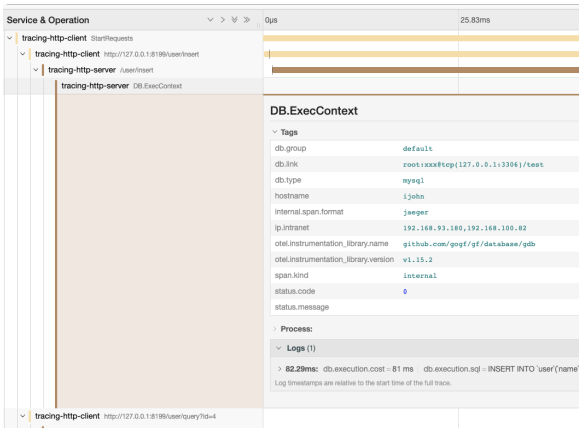


HTTP Client&ServerLoggingormredis

ORM

Attributes/Tags

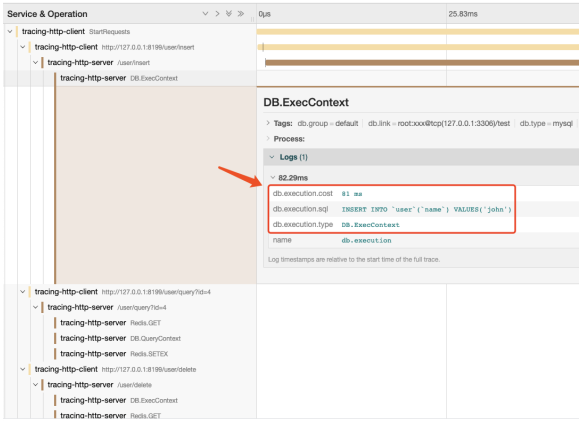
ORMSpanAttributes/Tags



`span.kindinternalspanTagsTags`

Attribute/Tag	
`db.type`	`mysql`, `mssql`, `pgsql`
`db.link`	
`db.group`	

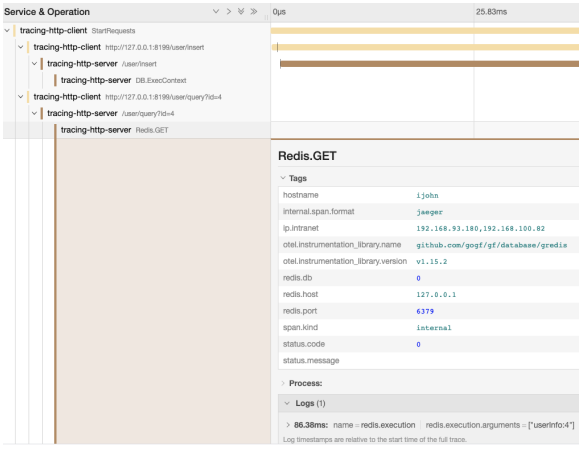
Events/Process



Event/Log	
`db.execution.sql`	`SQLORM`
`db.execution.type`	`SQLDB.ExecContextDB.QueryContext`
`db.execution.cost`	`SQLms`

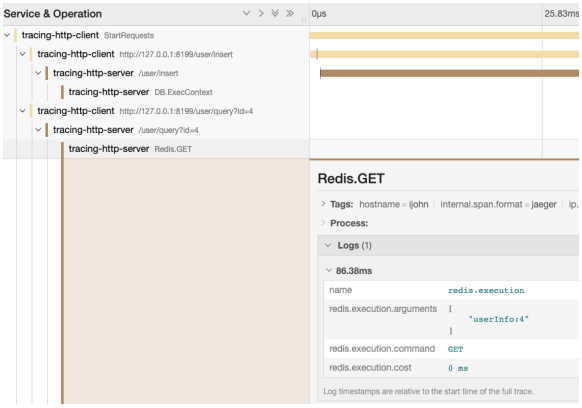
Redis

Attributes/Tags



Attribute/Tag	
redis.host	Redis
redis.port	Redis
redis.db	Redisdb

Events/Process



Event/Log	
redis.execution.command	Redis
redis.execution.arguments	Redis
redis.execution.cost	Redisms