

-TraceID

TraceIDTraceIDContextglogTraceIDglog

TraceID

1

GoFrameClientTraceIDGoFrameTraceIDOpenTelemetry32

 gclient

2

GoFrameServerTraceIDContextTraceID

TraceID

1

GoFrameClient

1TraceID

gctx.New/WithCtxTraceIDContextContextgctx.CtxIdTraceID

```
// New creates and returns a context which contains context id.
func New() context.Context

// WithCtx creates and returns a context containing context id upon given
parent context `ctx`.
func WithCtx(ctx context.Context) context.Context
```

WithCtxctxTraceIDTraceID

2TraceID

TraceIDgtrace.WithTraceID

```
// WithTraceID injects custom trace id into context to propagate.
func WithTraceID(ctx context.Context, traceID string) (context.Context,
error)
```

3Response Header

GoFrame Server Header Trace-Id

2

GoFrameServergctx.CtxIdTraceID

```
// CtxId retrieves and returns the context id from context.
func CtxId(ctx context.Context) string
```

Content Menu

-
- [TraceID](#)
 - 1
 - 2
- [TraceID](#)
 - 1
 - 1TraceID
 - 2TraceID
 - 3Response Header
 - 2
-
- 1HTTP Response Header TraceID
- 2TraceID
- 3TraceID
-
- 1GoFrameClient /ServerTraceID
- 2OpenTelemetry RequestID 33612a70-990a-11ea-87fe-fd68517e7a2d TraceID

1HTTP Response Header TraceID

```
package main

import (
    "context"
    "time"

    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/ghttp"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    s := g.Server()
    s.BindHandler("/", func(r *ghttp.Request) {
        traceID := gctx.CtxId(r.Context())
        g.Log().Info(r.Context(), "handler")
        r.Response.Write(traceID)
    })
    s.SetPort(8199)
    go s.Start()

    time.Sleep(time.Second)

    req, err := g.Client().Get(context.Background(), "http://127.0.0.1:8199/")
    if err != nil {
        panic(err)
    }
    defer req.Close()
    req.RawDump()
}
```

```

ADDRESS | METHOD | ROUTE |
HANDLER                                | MIDDLEWARE
-----|-----|-----
|-----|
|-----|
:8199   | ALL    | /      | main.main.
func1                                |
-----|-----|-----
|-----|
:8199   | ALL    | /*     | github.com/gogf/gf/v2/net/ghttp.
internalMiddlewareServerTracing | GLOBAL MIDDLEWARE
-----|-----|-----
|-----|
|-----|

2022-06-06 21:14:37.245 [INFO] pid[55899]: http server started listening
on [:8199]
2022-06-06 21:14:38.247 [INFO] {908d2027560af616e218e912d2ac3972} handler
+-----+
| REQUEST                                |
+-----+
GET / HTTP/1.1
Host: 127.0.0.1:8199
User-Agent: GClient v2.1.0-rc4 at TXQIANGGUO-MB0
Traceparent: 00-908d2027560af616e218e912d2ac3972-1e291041b9afa718-01
Accept-Encoding: gzip

+-----+
| RESPONSE                                |
+-----+
HTTP/1.1 200 OK
Connection: close
Content-Length: 32
Content-Type: text/plain; charset=utf-8
Date: Mon, 06 Jun 2022 13:14:38 GMT
Server: GoFrame HTTP Server
Trace-Id: 908d2027560af616e218e912d2ac3972

908d2027560af616e218e912d2ac3972

```

2TraceID

```

package main

import (
    "time"

    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/ghttp"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    s := g.Server()
    s.BindHandler("/", func(r *ghttp.Request) {
        traceID := gctx.CtxId(r.Context())
        g.Log().Info(r.Context(), "handler")
        r.Response.Write(traceID)
    })
    s.SetPort(8199)
    go s.Start()

    time.Sleep(time.Second)

    ctx := gctx.New()
    g.Log().Info(ctx, "request starts")
    content := g.Client().GetContent(ctx, "http://127.0.0.1:8199/")
    g.Log().Infof(ctx, "response: %s", content)
}

```

```

2022-06-06 21:17:17.447 [INFO] pid[56070]: http server started listening
on [:8199]

```

ADDRESS	METHOD	ROUTE	MIDDLEWARE
:8199	ALL	/	main.main. func1
:8199	ALL	/*	github.com/gogf/gf/v2/net/ghttp. internalMiddlewareServerTracing GLOBAL MIDDLEWARE

```

2022-06-06 21:17:18.450 [INFO] {e843f0737b0af616d8ed185d46ba65c5} request
starts
2022-06-06 21:17:18.451 [INFO] {e843f0737b0af616d8ed185d46ba65c5} handler
2022-06-06 21:17:18.451 [INFO] {e843f0737b0af616d8ed185d46ba65c5}
response: e843f0737b0af616d8ed185d46ba65c5

```

3TraceID

```

package main

import (
    "context"
    "time"

    "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"
    "github.com/gogf/gf/v2/text/gstr"
)

func main() {
    s := g.Server()
    s.BindHandler("/", func(r *ghttp.Request) {
        traceID := gctx.CtxId(r.Context())
        g.Log().Info(r.Context(), "handler")
        r.Response.Write(traceID)
    })
    s.SetPort(8199)
    go s.Start()

    time.Sleep(time.Second)

    ctx, _ := gtrace.WithTraceID(context.Background(), gstr.Repeat(
        "a", 32))
    g.Log().Info(ctx, "request starts")
    content := g.Client().GetContent(ctx, "http://127.0.0.1:8199/")
    g.Log().Infof(ctx, "response: %s", content)
}

```

```

  ADDRESS | METHOD | ROUTE |
HANDLER  |        |        |      MIDDLEWARE
-----|-----|-----|
|-----|
|-----|
|-----|
:8199    | ALL   | /      | main.main.
func1    |
-----|-----|-----|
|-----|
|-----|
:8199    | ALL   | /*     | github.com/gogf/gf/v2/net/ghttp.
internalMiddlewareServerTracing | GLOBAL MIDDLEWARE
-----|-----|-----|
|-----|
|-----|

2022-06-06 21:40:03.897 [INFO] pid[58435]: http server started listening
on [:8199]
2022-06-06 21:40:04.900 [INFO] {aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa} request
starts
2022-06-06 21:40:04.901 [INFO] {aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa} handler
2022-06-06 21:40:04.901 [INFO] {aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa}
response: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

```

1GoFrameClient/ServerTraceID

GoFrameClient/Server

Client/Server

2OpenTelemetryRequestID 33612a70-990a-11ea-87fe-fd68517e7a2dTraceID

RequestIDTraceIDUUIDUUIDTraceIDServerRequestIDTraceIDTraceIDContextContext

```
package main

import (
    "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"
)

func main() {
    //
    internalServer := g.Server("internalServer")
    internalServer.BindHandler("/", func(r *ghttp.Request) {
        traceID := gctx.CtxId(r.Context())
        g.Log().Info(r.Context(), "internalServer handler")
        r.Response.Write(traceID)
    })
    internalServer.SetPort(8199)
    go internalServer.Start()

    //
    // http://127.0.0.1:8299/?RequestID=33612a70-990a-11ea-87fe-
    fd68517e7a2d
    userSideServer := g.Server("userSideServer")
    userSideServer.Use(func(r *ghttp.Request) {
        requestID := r.Get("RequestID").String()
        if requestID != "" {
            newCtx, err := gtrace.WithUUID(r.Context(),
requestID)
            if err != nil {
                panic(err)
            }
            r.SetCtx(newCtx)
        }
        r.Middleware.Next()
    })
    userSideServer.BindHandler("/", func(r *ghttp.Request) {
        ctx := r.Context()
        g.Log().Info(ctx, "request internalServer starts")
        content := g.Client().GetContent(ctx, "http://127.0.0.1:
8199/")
        g.Log().Infof(ctx, "internalServer response: %s", content)
        r.Response.Write(content)
    })
    userSideServer.SetPort(8299)
    userSideServer.Run()
}
```

HTTP<http://127.0.0.1:8299/?RequestID=33612a70-990a-11ea-87fe-fd68517e7a2d>

2022-06-07 14:51:21.957 [INFO] openapi specification is disabled
2022-06-07 14:51:21.958 [INTE] ghttp_server.go:78 78198: graceful reload
feature is disabled

SERVER	DOMAIN	ADDRESS	METHOD	ROUTE	HANDLER
MIDDLEWARE					
internalServer	default	:8199	ALL	/	main.main.func1
internalServer	default	:8199	ALL	/*	github.com/gogf/gf/v2/net/ghttp.internalMiddlewareServerTracing GLOBAL MIDDLEWARE

2022-06-07 14:51:21.959 [INFO] pid[78198]: http server started listening on [:8199]
2022-06-07 14:51:21.965 [INFO] openapi specification is disabled

SERVER	DOMAIN	ADDRESS	METHOD	ROUTE	HANDLER
MIDDLEWARE					
userSideServer	default	:8299	ALL	/	main.main.func3
userSideServer	default	:8299	ALL	/*	github.com/gogf/gf/v2/net/ghttp.internalMiddlewareServerTracing GLOBAL MIDDLEWARE
userSideServer	default	:8299	ALL	/*	main.main.func2 GLOBAL MIDDLEWARE

2022-06-07 14:51:21.965 [INFO] pid[78198]: http server started listening on [:8299]
2022-06-07 14:53:05.322 [INFO] {33612a70990a11ea87fef68517e7a2d} request internalServer starts
2022-06-07 14:53:05.323 [INFO] {33612a70990a11ea87fef68517e7a2d} internalServer handler
2022-06-07 14:53:05.323 [INFO] {33612a70990a11ea87fef68517e7a2d} internalServer response: 33612a70990a11ea87fef68517e7a2d

RequestIDTraceID

