

GoFrameGRPCgrpcx<https://github.com/gogf/gf/tree/master/contrib/rpc/grpcx>

## -Server

grpcx.Server<https://github.com/gogf/gf/blob/master/example/rpc/grpcx/basic/server/main.go>

```
package main

import (
    "github.com/gogf/gf/contrib/rpc/grpcx/v2"
    "github.com/gogf/gf/example/rpc/grpcx/basic/controller"
)

func main() {
    s := grpcx.Server.New()
    controller.Register(s)
    s.Run()
}
```

### Content Menu

- [-Server](#)
- [-Client](#)
- [-Ctx](#)
- [-Balancer](#)
- [-Resolver](#)

## -Client

grpcx.Client<https://github.com/gogf/gf/blob/master/example/rpc/grpcx/basic/client/main.go>

```
package main

import (
    "github.com/gogf/gf/contrib/rpc/grpcx/v2"
    "github.com/gogf/gf/example/rpc/grpcx/basic/protobuf"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctxx"
)

func main() {
    var (
        ctx    = gctxx.New()
        conn   = grpcx.Client.MustNewGrpcClientConn("demo")
        client = protobuf.NewGreeterClient(conn)
    )
    res, err := client.SayHello(ctx, &protobuf.HelloRequest{Name:
"World"})
    if err != nil {
        g.Log().Error(ctx, err)
        return
    }
    g.Log().Debug(ctx, "Response:", res.Message)
}
```



### Client

grpc ClientClientGC

## -Ctx

grpcx.Ctx

```

func (c Ctx) NewIncoming(ctx context.Context, data ...g.Map) context.
Context
func (c Ctx) NewOutgoing(ctx context.Context, data ...g.Map) context.
Context
func (c Ctx) IncomingToOutgoing(ctx context.Context, keys ...string)
context.Context
func (c Ctx) IncomingMap(ctx context.Context) *gmap.Map
func (c Ctx) OutgoingMap(ctx context.Context) *gmap.Map
func (c Ctx) SetIncoming(ctx context.Context, data g.Map) context.Context
func (c Ctx) SetOutgoing(ctx context.Context, data g.Map) context.Context

```

OutgoingKey-ValueMapIncomingIncoming

server.go

```

package main

import (
    "github.com/gogf/gf/contrib/rpc/grpcx/v2"
    "github.com/gogf/gf/example/rpc/grpcx/ctx/controller"
)

func main() {
    s := grpcx.Server.New()
    controller.Register(s)
    s.Run()
}

```

controller.go

```

package controller

import (
    "context"

    "github.com/gogf/gf/contrib/rpc/grpcx/v2"
    "github.com/gogf/gf/example/rpc/grpcx/ctx/protobuf"
    "github.com/gogf/gf/v2/frame/g"
)

type Controller struct {
    protobuf.UnimplementedGreeterServer
}

func Register(s *grpcx.GrpcServer) {
    protobuf.RegisterGreeterServer(s.Server, &Controller{})
}

// SayHello implements helloworld.GreeterServer
func (s *Controller) SayHello(ctx context.Context, in *protobuf.
HelloRequest) (*protobuf.HelloReply, error) {
    m := grpcx.Ctx.IncomingMap(ctx)
    g.Log().Infof(ctx, `incoming data: %v`, m.Map())
    return &protobuf.HelloReply{Message: "Hello " + in.GetName()}, nil
}

```

client.go

```

package main

import (
    "github.com/gogf/gf/contrib/rpc/grpcx/v2"
    "github.com/gogf/gf/example/rpc/grpcx/ctx/protobuf"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    var (
        conn    = grpcx.Client.MustNewGrpcClientConn("demo")
        client = protobuf.NewGreeterClient(conn)
        ctx     = grpcx.Ctx.NewOutgoing(gctx.New(), g.Map{
            "UserId":    "1000",
            "UserName": "john",
        })
    )
    g.Log().Infof(ctx, `outgoing data: %v`, grpcx.Ctx.OutgoingMap(ctx).
Map())
    res, err := client.SayHello(ctx, &protobuf.HelloRequest{Name:
"World"})
    if err != nil {
        g.Log().Error(ctx, err)
        return
    }
    g.Log().Debug(ctx, "Response:", res.Message)
}

```

```

$ go run server.go
2023-03-15 19:28:45.331 [DEBU] set default registry using file registry as
no custom registry set
2023-03-15 19:28:45.331 [DEBU] service register: &{Head: Deployment:
Namespace: Name:demo Version: Endpoints:10.35.12.81:51707 Metadata:map
[protocol:grpc]}
2023-03-15 19:28:45.332 [INFO] pid[83753]: grpc server started listening
on [:51707]
2023-03-15 19:28:54.093 [INFO] {d049db1a39944c1711bd9f37d58a88f9} incoming
data: map[:authority:service/default/default/demo/latest/ content-type:
application/grpc traceparent:00-d049db1a39944c1711bd9f37d58a88f9-
adbd2ba657ffea45-01 user-agent:grpc-go/1.49.0 userid:1000 username:john]
2023-03-15 19:28:54.093 {d049db1a39944c1711bd9f37d58a88f9} /protobuf.
Greeter/SayHello, 0.049ms, name:"World", message:"Hello World"

```

```

$ go run client.go
2023-03-15 19:28:54.087 [INFO] {d049db1a39944c1711bd9f37d58a88f9} outgoing
data: map[userid:1000 username:john]
2023-03-15 19:28:54.089 [DEBU] client conn updated with addresses
[{"Addr":"10.35.12.81:51707","ServerName":"demo","Attributes":{},"
BalancerAttributes":null,"Type":0,"Metadata":null}]
2023-03-15 19:28:54.093 [DEBU] {d049db1a39944c1711bd9f37d58a88f9}
Response: Hello World

```

## -Balancer

grpcx.Balancer

server.go

```

package main

import (
    "github.com/gogf/gf/contrib/rpc/grpcx/v2"
    "github.com/gogf/gf/example/rpc/grpcx/balancer/controller"
)

func main() {
    s := grpcx.Server.New()
    controller.Register(s)
    s.Run()
}

```

client.go

```

package main

import (
    "context"

    "github.com/gogf/gf/contrib/rpc/grpcx/v2"
    "github.com/gogf/gf/example/rpc/grpcx/balancer/protobuf"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    var (
        ctx    context.Context
        conn   = grpcx.Client.MustNewGrpcClientConn("demo", grpcx.
Balancer.WithRandom())
        client = protobuf.NewGreeterClient(conn)
    )
    for i := 0; i < 10; i++ {
        ctx = gctx.New()
        res, err := client.SayHello(ctx, &protobuf.HelloRequest
{Name: "World"})
        if err != nil {
            g.Log().Error(ctx, err)
            return
        }
        g.Log().Debug(ctx, "Response:", res.Message)
    }
}

```

grpcx.Balancer.WithRandom()

server.goclient.go

server1

```
$ go run server.go
2023-03-15 19:50:44.801 [DEBU] set default registry using file registry as
no custom registry set
2023-03-15 19:50:44.802 [DEBU] service register: &{Head: Deployment:
Namespace: Name:demo Version: Endpoints:10.35.12.81:53962 Metadata:map
[protocol:grpc]}
2023-03-15 19:50:44.802 [INFO] pid[89290]: grpc server started listening
on [:53962]
2023-03-15 19:50:57.282 {7025612f6d954c17c5f335051bf10899} /protobuf.
Greeter/SayHello, 0.003ms, name:"World", message:"Hello World"
2023-03-15 19:50:57.283 {60567c2f6d954c17c7f335052ce05185} /protobuf.
Greeter/SayHello, 0.002ms, name:"World", message:"Hello World"
2023-03-15 19:50:57.285 {f8d09b2f6d954c17ccf33505dff1a4ea} /protobuf.
Greeter/SayHello, 0.002ms, name:"World", message:"Hello World"
2023-03-15 19:50:57.287 {f0fab02f6d954c17cdf33505438b2c80} /protobuf.
Greeter/SayHello, 0.001ms, name:"World", message:"Hello World"
```

server2

```
$ go run server.go
2023-03-15 19:50:51.720 [DEBU] set default registry using file registry as
no custom registry set
2023-03-15 19:50:51.721 [DEBU] service register: &{Head: Deployment:
Namespace: Name:demo Version: Endpoints:10.35.12.81:53973 Metadata:map
[protocol:grpc]}
2023-03-15 19:50:51.722 [INFO] pid[89351]: grpc server started listening
on [:53973]
2023-03-15 19:50:57.280 {b89a0d2f6d954c17c4f33505a046817c} /protobuf.
Greeter/SayHello, 0.002ms, name:"World", message:"Hello World"
2023-03-15 19:50:57.282 {28bf732f6d954c17c6f33505adedff5f} /protobuf.
Greeter/SayHello, 0.002ms, name:"World", message:"Hello World"
2023-03-15 19:50:57.283 {9876832f6d954c17c8f3350580ed535b} /protobuf.
Greeter/SayHello, 0.002ms, name:"World", message:"Hello World"
2023-03-15 19:50:57.284 {684e8b2f6d954c17c9f33505d56e4b05} /protobuf.
Greeter/SayHello, 0.001ms, name:"World", message:"Hello World"
2023-03-15 19:50:57.284 {c045912f6d954c17caf3350599006197} /protobuf.
Greeter/SayHello, 0.001ms, name:"World", message:"Hello World"
2023-03-15 19:50:57.284 {500a972f6d954c17cbf33505252b0e01} /protobuf.
Greeter/SayHello, 0.001ms, name:"World", message:"Hello World"
```

```

$ go run client.go
2023-03-15 19:50:57.278 [DEBU] client conn updated with addresses
[{"Addr":"10.35.12.81:53962","ServerName":"demo","Attributes":{},"BalancerAttributes":null,"Type":0,"Metadata":null}, {"Addr":"10.35.12.81:53973","ServerName":"demo","Attributes":{},"BalancerAttributes":null,"Type":0,"Metadata":null}]
2023-03-15 19:50:57.281 [DEBU] {b89a0d2f6d954c17c4f33505a046817c}
Response: Hello World
2023-03-15 19:50:57.282 [DEBU] {7025612f6d954c17c5f335051bf10899}
Response: Hello World
2023-03-15 19:50:57.282 [DEBU] {28bf732f6d954c17c6f33505adedff5f}
Response: Hello World
2023-03-15 19:50:57.283 [DEBU] {60567c2f6d954c17c7f335052ce05185}
Response: Hello World
2023-03-15 19:50:57.283 [DEBU] {9876832f6d954c17c8f3350580ed535b}
Response: Hello World
2023-03-15 19:50:57.284 [DEBU] {684e8b2f6d954c17c9f33505d56e4b05}
Response: Hello World
2023-03-15 19:50:57.284 [DEBU] {c045912f6d954c17caf3350599006197}
Response: Hello World
2023-03-15 19:50:57.285 [DEBU] {500a972f6d954c17cbf33505252b0e01}
Response: Hello World
2023-03-15 19:50:57.286 [DEBU] {f8d09b2f6d954c17ccf33505dff1a4ea}
Response: Hello World
2023-03-15 19:50:57.287 [DEBU] {f0fab02f6d954c17cdf33505438b2c80}
Response: Hello World

```

1046

## -Resolver

grpcx.ResolverGRPCgrpcx

server.go

```

package main

import (
    "github.com/gogf/gf/contrib/registry/etcd/v2"
    "github.com/gogf/gf/contrib/rpc/grpcx/v2"
    "github.com/gogf/gf/example/rpc/grpcx/resolver/controller"
)

func main() {
    grpcx.Resolver.Register(etcd.New("127.0.0.1:2379"))

    s := grpcx.Server.New()
    controller.Register(s)
    s.Run()
}

```

grpcx.Resolver.Register(etcd.New("127.0.0.1:2379")) etcdGRPC

client.go

```

package main

import (
    "github.com/gogf/gf/contrib/registry/etcd/v2"
    "github.com/gogf/gf/contrib/rpc/grpcx/v2"
    "github.com/gogf/gf/example/rpc/grpcx/resolver/protobuf"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    grpcx.Resolver.Register(etcd.New("127.0.0.1:2379"))

    var (
        ctx    = gctx.New()
        conn   = grpcx.Client.MustNewGrpcClientConn("demo")
        client = protobuf.NewGreeterClient(conn)
    )
    res, err := client.SayHello(ctx, &protobuf.HelloRequest{Name:
"World"})
    if err != nil {
        g.Log().Error(ctx, err)
        return
    }
    g.Log().Debug(ctx, "Response:", res.Message)
}

```

etcd

```

$ etcd
{"level":"info","ts":"2023-03-15T20:02:50.966+0800","caller":"etcdmain
/etcd.go:73","msg":"Running: ", "args":["etcd"]}
{"level":"info","ts":"2023-03-15T20:02:50.967+0800","caller":"etcdmain
/etcd.go:100","msg":"failed to detect default host","error":"default host
not supported on darwin_amd64"}
{"level":"warn","ts":"2023-03-15T20:02:50.967+0800","caller":"etcdmain
/etcd.go:105","msg":"'data-dir' was empty; using default","data-dir":"
default.etcd"}
{"level":"info","ts":"2023-03-15T20:02:50.967+0800","caller":"etcdmain
/etcd.go:116","msg":"server has been already initialized","data-dir":"
default.etcd","dir-type":"member"}
{"level":"info","ts":"2023-03-15T20:02:50.967+0800","caller":"embed/etcd.
go:124","msg":"configuring peer listeners","listen-peer-urls":
["http://localhost:2380"]}
{"level":"info","ts":"2023-03-15T20:02:50.968+0800","caller":"embed/etcd.
go:132","msg":"configuring client listeners","listen-client-urls":
["http://localhost:2379"]}
...

```

server.go

```

$ go run server.go
2023-03-15 20:02:19.472 [DEBU] service register: &{Head: Deployment:
Namespace: Name:demo Version: Endpoints:10.35.12.81:55066 Metadata:map
[protocol:grpc]}
2023-03-15 20:02:19.516 [DEBU] etcd put success with key "/service/default
/default/demo/latest/10.35.12.81:55066", value "{ \"protocol\": \"grpc\" }",
lease "7587869265929863945"
2023-03-15 20:02:19.516 [INFO] pid[92040]: grpc server started listening
on [:55066]
2023-03-15 20:02:29.310 [88c4c04e0e964c17dddec53bladb54f7] /protobuf.
Greeter/SayHello, 0.001ms, name:"World", message:"Hello World"

```

client.go

```
$ go run client.go
2023-03-15 20:02:29.297 [DEBU] Watch key "/service/default/default/demo
/latest/"
2023-03-15 20:02:29.307 [DEBU] client conn updated with addresses
[{"Addr":"10.35.12.81:55066","ServerName":"demo","Attributes":{},"
BalancerAttributes":null,"Type":0,"Metadata":null}]
2023-03-15 20:02:29.308 [DEBU] client conn updated with addresses
[{"Addr":"10.35.12.81:55066","ServerName":"demo","Attributes":{},"
BalancerAttributes":null,"Type":0,"Metadata":null}]
2023-03-15 20:02:29.310 [DEBU] {88c4c04e0e964c17dddec53bladb54f7}
Response: Hello World
```