

GoFramegsvc <https://github.com/gogf/gf/tree/master/contrib/registry> etcd, zookeeper, polarisgsvc

etcd

```
package main

import (
    "github.com/gogf/gf/contrib/registry/etcd/v2"
    "github.com/gogf/gf/v2/net/gsvc"
)

func main() {
    gsvc.SetRegistry(etcd.New(`127.0.0.1:2379`))

    // ...
}
```

file	https://github.com/gogf/gf/tree/master/contrib/registry/file	
etcd	https://github.com/gogf/gf/tree/master/contrib/registry/etcd	
polaris	https://github.com/gogf/gf/tree/master/contrib/registry/polaris	
zookeeper	https://github.com/gogf/gf/tree/master/contrib/registry/zookeeper	

<https://github.com/gogf/gf/tree/master/contrib/registry>

HTTP

```
gsvc.SetRegistry(etcd.New(`127.0.0.1:2379`))etcdetcd.Newgsvc.Registrygsvc.
SetRegistry

server.go
```

Content Menu

-
-
-
-
- - [HTTP](#)
 - [GRPC](#)
-

```

package main

import (
    "github.com/gogf/gf/contrib/registry/etcd/v2"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/ghttp"
    "github.com/gogf/gf/v2/net/gsvc"
)

func main() {
    gsvc.SetRegistry(etcd.New(`127.0.0.1:2379`))

    s := g.Server(`hello.svc`)
    s.BindHandler("/", func(r *ghttp.Request) {
        g.Log().Info(r.Context(), `request received`)
        r.Response.Write(`Hello world`)
    })
    s.Run()
}

```

client.go

```

package main

import (
    "github.com/gogf/gf/contrib/registry/etcd/v2"
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/net/gsvc"
    "github.com/gogf/gf/v2/os/gctx"
)

func main() {
    gsvc.SetRegistry(etcd.New(`127.0.0.1:2379`))
    ctx := gctx.New()
    res := g.Client().GetContent(ctx, `http://hello.svc/`)
    g.Log().Info(ctx, res)
}

```

```
$ go run server.go
2023-03-15 20:55:56.256 [INFO] pid[3358]: http server started listening on
[:60700]
2023-03-15 20:55:56.256 [INFO] openapi specification is disabled
2023-03-15 20:55:56.256 [DEBU] service register: &{Head: Deployment:
Namespace: Name:hello.svc Version: Endpoints:10.35.12.81:60700 Metadata:map
[insecure:true protocol:http]}
2023-03-15 20:55:56.297 [DEBU] etcd put success with key "/service/default
/default/hello.svc/latest/10.35.12.81:60700", value "{\"insecure\":true,\"
protocol\":\"http\"}", lease "7587869265945813002"

  SERVER   | DOMAIN   | ADDRESS | METHOD | ROUTE
  |                               |
  MIDDLEWARE
  -----|-----|-----|-----|-----
  |-----|-----|-----|-----|-----
  |-----|-----|-----|-----|-----
  |-----|-----|-----|-----|-----
  hello.svc | default | :60700  | ALL   | /      | main.main.
func1      |         |         |         |         |
  -----|-----|-----|-----|-----
  |-----|-----|-----|-----|-----
  |-----|-----|-----|-----|-----
  hello.svc | default | :60700  | ALL   | /*     | github.com/gogf/gf/v2
/net/ghttp.internalMiddlewareServerTracing | GLOBAL MIDDLEWARE
  -----|-----|-----|-----|-----
  |-----|-----|-----|-----|-----
  |-----|-----|-----|-----|-----

2023-03-15 20:56:45.739 [INFO] {880eaa8104994c17ffb384495cd4c613} request
received
```

```
$ go run client.go
2023-03-15 20:56:45.739 [INFO] {880eaa8104994c17ffb384495cd4c613} Hello
world
```

GRPC



GRPCgrpcx.ResolvergrpcxServer config.yaml grpc.name default

server.go

etcd.Newgsvc.Registrygrpcx.Resolver.Registergrpc

```
package main

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

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

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

config.yaml

```
grpc:
  name:          "demo" #
  address:       ":8000" #
  logPath:       "./log" #
  logStdout:     true   #
  errorLogEnabled: true  #
  accessLogEnabled: true #
  errorStack:    true   #
```

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/registry/etcd/grpc/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)
}
```

```
$ go run server.go
2023-03-15 21:06:57.204 [DEBU] service register: &{Head: Deployment:
Namespace: Name:demo Version: Endpoints:10.35.12.81:61978 Metadata:map
[protocol:grpc]}
2023-03-15 21:06:57.257 [DEBU] etcd put success with key "/service/default
/default/demo/latest/10.35.12.81:61978", value "{\"protocol\":\"grpc\"}",
lease "7587869265945813015"
2023-03-15 21:06:57.257 [INFO] pid[5786]: grpc server started listening on
[:61978]
2023-03-15 21:07:04.955 {08f0aead94994c1731591d2b653ddc18} /protobuf.
Greeter/SayHello, 0.002ms, name:"World", message:"Hello World"
```

```
$ go run client.go
2023-03-15 21:07:04.950 [DEBU] Watch key "/service/default/default/demo
/latest/"
2023-03-15 21:07:04.952 [DEBU] client conn updated with addresses
[{"Addr":"10.35.12.81:61978","ServerName":"demo","Attributes":{},"
BalancerAttributes":null,"Type":0,"Metadata":null}]
2023-03-15 21:07:04.953 [DEBU] client conn updated with addresses
[{"Addr":"10.35.12.81:61978","ServerName":"demo","Attributes":{},"
BalancerAttributes":null,"Type":0,"Metadata":null}]
2023-03-15 21:07:04.955 [DEBU] {08f0aead94994c1731591d2b653ddc18}
Response: Hello World
```

```
gclientgclientIP:PORT
```

```
gclientgclient.ClientDiscovery(nil)SetDiscovery(nil)
```

```
//
g.Client().Discovery(nil).Get(ctx, "http://192.168.1.1/api/v1/user")

//
client := g.Client()
client.SetDiscovery(nil)
client.Get(ctx, "http://192.168.1.1/api/v1/user")
```