**测试用例**

**1. 测试用例**

1.1 AServiceName测试用例

此文件路径为./rpcsvr/main\_test.go

const (

exampleMethodURL = "http://127.0.0.1:8888/AServiceName/ExampleMethod"

)

var httpCli = &http.Client{Timeout: 3 \* time.Second}

func BenchmarkAServiceName(b \*testing.B) {

for i := 1; i < b.N; i++ {

newExample := genExample(i)

resp, err := example\_(newExample)

Assert(b, err == nil, err)

fmt.Println(\*resp)

}

}

func example\_(exm \*server.ExampleReq) (rResp \*string, err error) {

reqBody, err := json.Marshal(exm)

if err != nil {

return nil, fmt.Errorf("marshal request failed: err=%v", err)

}

reader := bytes.NewReader(reqBody)

req, err := http.NewRequest("POST", exampleMethodURL, reader)

defer req.Body.Close()

req.Header.Set("Content-Type", "application/json;charset=UTF-8")

var resp \*http.Response

resp, err = httpCli.Do(req)

defer resp.Body.Close()

if err != nil {

return

}

var body []byte

if body, err = ioutil.ReadAll(resp.Body); err != nil {

return

}

if err = json.Unmarshal(body, &rResp); err != nil {

return

}

return

}

func genExample(id int) \*server.ExampleReq {

return &server.ExampleReq{

Msg: fmt.Sprintf("test-%d", id),

Base: &base.Base{

LogID: fmt.Sprint(id),

Caller: "czt",

Addr: "163",

Client: "c",

TrafficEnv: &base.TrafficEnv{

Open: true,

Env: "env",

},

Extra: nil,

},

}

}

// Assert asserts cond is true, otherwise fails the test.

func Assert(t testingTB, cond bool, val ...interface{}) {

t.Helper()

if !cond {

if len(val) > 0 {

val = append([]interface{}{"assertion failed:"}, val...)

t.Fatal(val...)

} else {

t.Fatal("assertion failed")

}

}

}

// testingTB is a subset of common methods between \*testing.T and \*testing.B.

type testingTB interface {

Fatal(args ...interface{})

Fatalf(format string, args ...interface{})

Helper()

}

1.2 idlSvr测试用例

此文件路径为./idlsvr/main\_test.go

const (

addIdlURL = "http://127.0.0.1:8888/idl-info/add/CServiceName"

)

var httpCli = &http.Client{Timeout: 3 \* time.Second}

func BenchmarkAServiceName(b \*testing.B) {

for i := 1; i < b.N; i++ {

newIdl := genExample(i)

resp, err := register\_(newIdl)

Assert(b, err == nil, err)

fmt.Println(\*resp)

}

}

func register\_(idl \*demo.IdlReq) (rResp \*string, err error) {

reqBody, err := json.Marshal(idl)

if err != nil {

return nil, fmt.Errorf("marshal request failed: err=%v", err)

}

reader := bytes.NewReader(reqBody)

req, err := http.NewRequest("POST", addIdlURL, reader)

defer req.Body.Close()

req.Header.Set("Content-Type", "application/json;charset=UTF-8")

var resp \*http.Response

resp, err = httpCli.Do(req)

defer resp.Body.Close()

if err != nil {

return

}

var body []byte

if body, err = ioutil.ReadAll(resp.Body); err != nil {

return

}

if err = json.Unmarshal(body, &rResp); err != nil {

return

}

return

}

func genExample(id int) \*demo.IdlReq {

return &demo.IdlReq{

IdlName: fmt.Sprintf("idl-test-%d", id),

IdlInfo: &demo.IdlInfo{

Content: fmt.Sprint(id),

Includes: nil,

},

}

}

// Assert asserts cond is true, otherwise fails the test.

func Assert(t testingTB, cond bool, val ...interface{}) {

t.Helper()

if !cond {

if len(val) > 0 {

val = append([]interface{}{"assertion failed:"}, val...)

t.Fatal(val...)

} else {

t.Fatal("assertion failed")

}

}

}

// testingTB is a subset of common methods between \*testing.T and \*testing.B.

type testingTB interface {

Fatal(args ...interface{})

Fatalf(format string, args ...interface{})

Helper()

}

总结：经过上述两个测试，表明API网关已接通，rpc服务和面向外部的添加idl的服务都能正常运行