

golang unit testing

单元测试主要目的是为了测试自己写的代码的功能是否符合自己的预期，优点在于如果将单元测试维护好的情况下，代码中每次有修改，都可以使用单元测试验证，一劳永逸。

注意事项

- 文件名必须是_test.go结尾的，这样在执行go test的时候才会执行到相应的代码
- 你必须import testing这个包
- 所有的测试用例函数必须是Test开头
- 测试用例会按照源代码中写的顺序依次执行
- 测试函数TestXxx()的参数是testing.T，我们可以使用该类型来记录错误或者是测试状态
- 函数中通过调用testing.T的Error, Errorf, FailNow, Fatal, Fatallf方法，说明测试不通过，调用Log方法用来记录测试的信息。

```
func TestCreateResourceTag(t *testing.T) {
    cli, err := getPermissionClient()
    if err != nil {
        t.Error("get cli conn failed")
        return
    }

    req := &permission.CreateResourceTag_Request{
        Domain:      domain,
        ResourceTag: resourceTag,
    }
    resq, err := cli.CreateResourceTag(context.Background(), req)
    if err != nil {
        t.Error("CreateResourceTag error:", err)
        return
    }
    if resq.Domain != domain || resq.Category != category || resq.TagName
    != tagName || resq.CallbackUrl != callbackUrl ||
        resq.CallbackType != callbackType {
        t.Error("接口调通，数据不正确")
        return
    }
    t.Log("CreateResourceTag success")
    return
}
```

执行测试文件进行测试

- go test //执行所有的测试文件
- go test XXX_test.go //执行单个文件

- `go test XXX_test.go -test.run TestXXX` 方法 //测试一个文件中的单个方法
- `go test -v XXX_test.go` // 加上-v 参数可以输出测试的具体细节

运行wed接口测试时遇见的问题

- 测试时连接不通 返回403 禁止访问code, 需要关闭代理, `unset http_proxy,unset https_proxy`
`unalias go` //具体别名