

课程安排，请关注微信公众平台或者官方微博

编程语言： **Golang** 与 **html5**

编程工具： **Goland** 和 **HBuilder**

预计平均一周左右更新一或二节课程

授人以鱼，不如授人以渔。

大家好，我是彬哥，

欢迎大家来到 **Golang** 语言社区云课堂课程的学习。

社区论坛网址：www.golang.ltd

技术交流群： **221273219**

微信公众号： **Golang 语言社区**

微信服务号： **Golang 技术社区**

第一季 Go 语言基础、进阶、提高课程

第十八节 Go 语言 接口

go 语言不是传统的面向对象语言，因此没有类和继承的概念。

go 语言中的接口跟 **java** 中的接口有点类似，接口可以有多个实现。

go 语言的接口是 **go** 的一种类型，用来指定一组方法(方法集)

go 的接口归根结底就是一个方法集合。

在 **go** 语言中，的接口有下面几个特点：

- 可以包含 0 个或多个方法的签名
- 只定义方法的签名，不包含实现
- 实现接口不需要显式的声明，只需实现相应方法即可

接口的定义

定义方式如下：

```
type Namer interface {  
  
    method1(param_list) return_list
```

```
method2(param_list) return_list

...

}
```

go 语言中的接口往往都很短，不会存在太多方法，一般只有 0 到 5 个。

开始实战：

定义：

```
Type IProxy interface{

    GET ()

    POST ()

}
```

实现一个接口：

Go 的 **interface** 是一种内置类型，属于动态风格的 **duck-typing** 类型。接口作为方法签名的集合，任何类型的方法集中只要拥有与之对应的全部方法，就表示它实现了该接口。

```
Type Proxy struct {}

Func(this *Proxy) GET () {

}

Func(this *Proxy) POST {

}
```

接口调用：

```
Var var_proxy IProxy

var_proxy = new(Proxy)

var_proxy.GET ()

var_proxy.POST ()
```

知识拓展—接口的使用

- 1 作为函数参数
- 2 作为参数返回值
- 3 嵌套接口

interface 底层结构

interface 是一个结构体，包含两个成员。根据 interface 是否包含方法，底层又分为两个结构体。eface 主要是保存了类型信息，以后总结反射时具体讲，这里先总结带方法的 iface。结构体定义在 runtime2.go 显然 iface 由两部分组成，data 域保存元数据，tab 描述接口。

```
type eface struct {
    _type *_type
    data unsafe.Pointer
}
type iface struct {
    tab *itab
    data unsafe.Pointer
}
```

```
type itab struct {
    inter *interfacetype // 保存该接口的方法签名
    _type *_type // 保存动态类型的type类型信息
    link *itab // 可能有嵌套的itab
    bad int32
    unused int32
    fun [1]uintptr // 保存动态类型对应的实现
}
type interfacetype struct {
    type _type
    mhdr []imethod
}
```



成都字节跳动