# 适配器模式

1. **基本概念**

### **1、定义：**

适配器（Adapter）模式的定义：适配器模式，将一个类的基础上进行封装，封装的接口转化成客户希望的另外一个接口。Adapter模式使得原本不兼容而不能一起工作的那些类可以一起工作。

简单来说，适配器模式就是让两个不同接口想要相互使用的时候，进行多嵌套一层。

一般在设计接近尾声的时候想要进行接口调用进行使用，或者是时间来不及的时候进行使用比较方便，当然最好的办法是重构接口，这样会让代码看起来更加美观，但是适配器模式的存在可以最大程度上以最小的代价去弥补设计的缺陷。

### 2、组成环节：

目标类（Target）：Target类里面所存的数据就是客户所期待的接口，可以是具体或者抽象的类。

适配器类（Adapter）：通过在内部包装一个Adaptee对象，将源接口转化成目标接口。

源类（Adaptee）：这就是需要去被适配的类。

## 二、go语言中实现

##### 1、首先定义一个Target的目标类

type target interface {  
 request()  
}

##### 2、创建adaptee的源类，并进行方法的实现

type adaptee struct{}  
  
func (a adaptee) otherrequest() {  
 fmt.Println("This is otherrequest.")  
}

##### 3、创建一个adapter类，并进行request方法的实现

type adapter struct{}  
  
func (a adapter) request() {  
 var adp adaptee  
 adp.otherrequest()  
}

##### 4、通过main函数实现

func main() {  
 var a adapter  
 a.request()  
}

##### 5、输出结果

**This is otherrequest.**