

Interfaces in Go

Define the interface

```
type Greeter interface {  
    Greet(username string)  
}
```

Create an Object(struct and functions) that will implement the interface.

```
type NiceGreeter struct {}  
  
func (g NiceGreeter) Greet(username string){  
    fmt.Printf("Hi %s ",username)  
}
```

In the above, we can see that we simply implement the function of the interface.

We link the type (Object) to the function using **(db DefaultDatabase)** in the above code.

Use the interface

Define an Object(struct) that uses the interface

```
type Program struct {  
    Greeter Greeter  
}  
  
func (p Program) Execute(user string){  
    p.Greeter.Greet(user)  
}
```

```
func main(){  
  
    //instantiate the greeter that implements Greeter  
    greeter := NiceGreeter{}  
  
    //instantiate the program with our greeter  
    p:= Program{  
        Greeter: greeter,  
    }  
  
    //execute  
    p.Execute("bob")  
}
```

Full Code Example

```
package main

import "fmt"

type Greeter interface {
    Greet(username string)
}

type NiceGreeter struct {}

func (g NiceGreeter) Greet(username string){
    fmt.Printf("Hi %s ",username)
}

type Program struct {
    Greeter Greeter
}

func (p Program) Execute(user string){
    p.Greeter.Greet(user)
}

func main(){

    greeter := NiceGreeter{}
    p:= Program{
        Greeter: greeter,
    }
    p.Execute("bob")
}
```

References

Reference	URL
Interfaces - Go By Example	https://gobyexample.com/interfaces
Interfaces in Go	https://medium.com/rungo/interfaces-in-go-ab1601159b3a