

## GABRIELE ROMANATO

### Ordinare una slice di struct in Go

In Go, l'ordinamento di una slice contenente struct può essere effettuato utilizzando il pacchetto `sort`. Questo pacchetto fornisce la funzione `sort.Slice`, che consente di definire un criterio personalizzato per l'ordinamento.

### Esempio di Ordinamento

Consideriamo una struct `Person` con i campi `Name` e `Age`. L'obiettivo è ordinare una slice di `Person` in base all'età.

```
package main

import (
    "fmt"
    "sort"
)

type Person struct {
    Name string
    Age  int
}

func main() {
    people := []Person{
        {Name: "Alice", Age: 30},
        {Name: "Bob", Age: 25},
        {Name: "Charlie", Age: 35},
    }

    sort.Slice(people, func(i, j int) bool {
        return people[i].Age < people[j].Age
    })

    fmt.Println("Slice ordinata:", people)
}
```

### Spiegazione

- `sort.Slice` accetta la slice e una funzione di confronto che stabilisce l'ordine.
- La funzione di confronto restituisce `true` se l'elemento in posizione `i` deve precedere quello in posizione `j`.
- Nell'esempio, la slice viene ordinata in ordine crescente rispetto all'età.

### Ordinamento Personalizzato

Se si desidera un ordinamento più complesso, è possibile implementare l'interfaccia `sort.Interface`:

```
type ByName []Person

func (p ByName) Len() int           { return len(p) }
func (p ByName) Swap(i, j int)      { p[i], p[j] = p[j], p[i] }
func (p ByName) Less(i, j int) bool { return p[i].Name < p[j].Name }

func main() {
    people := []Person{
        {Name: "Alice", Age: 30},
    }
```

```
        {Name: "Bob", Age: 25},  
        {Name: "Charlie", Age: 35},  
    }  
  
    sort.Sort(ByName(people))  
  
    fmt.Println("Slice ordinata per nome:", people)  
}
```

Questo approccio è utile quando si vuole riutilizzare il criterio di ordinamento in più punti del codice.

## Conclusione

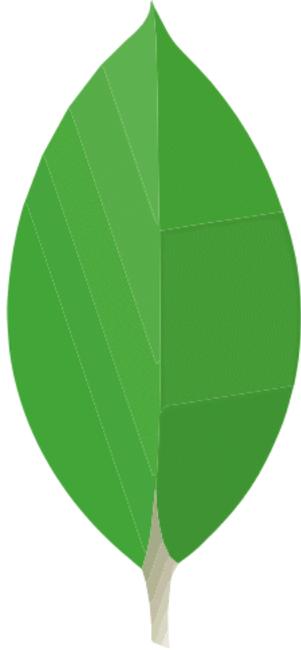
Go offre metodi semplici ed efficaci per ordinare slice di struct. `sort.Slice` è utile per ordinamenti rapidi e personalizzabili, mentre `sort.Interface` consente una maggiore riusabilità e flessibilità.

## Applicazioni Correlate



## Go Placeholder Image

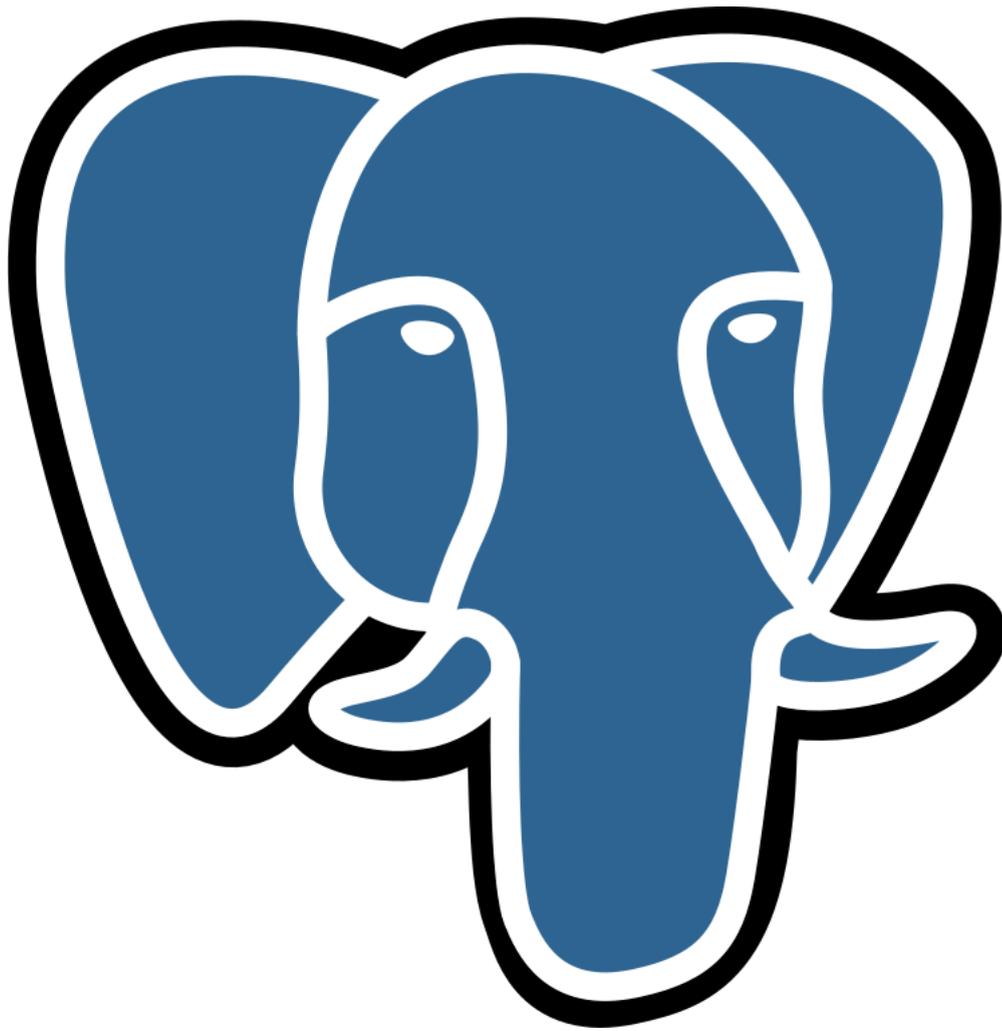
Applicazione in Go per la creazione di immagini segnaposto.  
Docker Docker Compose Go JavaScript



•

## Go MongoDB App

Applicazione basata su MongoDB ed implementata in Go con il driver ufficiale.  
Docker Docker Compose Go MongoDB



.

### **Go PostgreSQL App**

Applicazione basata su PostgreSQL e sviluppata in Go.  
Docker Docker Compose Go PostgreSQL