

GABRIELE ROMANATO

Introduzione ai generics in Go

I generics sono una delle funzionalità più attese introdotte in Go a partire dalla versione 1.18. Consentono di scrivere funzioni, metodi e strutture dati che funzionano con diversi tipi senza dover duplicare il codice. Questo rende il codice più riutilizzabile, sicuro e manutenibile.

Perché Usare i Generics?

Prima dei generics, era comune utilizzare il tipo `interface{}` per ottenere una sorta di "tipizzazione generica", ma questo approccio richiedeva dei cast espliciti e poteva introdurre errori a runtime. I generics permettono invece di definire il tipo in modo sicuro e controllato a compile time.

Come Funzionano i Generics in Go

In Go, i generics si basano sui **parametri di tipo**. Si definiscono tra parentesi quadre dopo il nome della funzione o del tipo.

```
func PrintSlice[T any](s []T) {
    for _, v := range s {
        fmt.Println(v)
    }
}
```

In questo esempio, `T` è un parametro di tipo che rappresenta qualsiasi tipo (`any` è un alias per `interface{}`).

Vincoli sui Tipi

È possibile imporre dei vincoli sui tipi usando le **constraint**. Ad esempio, possiamo limitare i tipi ai soli numerici:

```
type Number interface {
    int | float64
}

func Sum[T Number](a, b T) T {
    return a + b
}
```

In questo caso, `T` può essere solo `int` o `float64`.

Generics e Strutture

I generics possono essere utilizzati anche con le strutture:

```
type Pair[T any, U any] struct {
    First T
    Second U
}
```

Con questa definizione possiamo creare una coppia di qualsiasi tipo, ad esempio `Pair[int, string]`.

Considerazioni Finali

I generics rendono il linguaggio Go più potente senza comprometterne la semplicità. Tuttavia, è importante usarli con buon senso per mantenere il codice chiaro e leggibile. Go incoraggia uno stile pragmatico e l'introduzione dei generics segue questa filosofia.

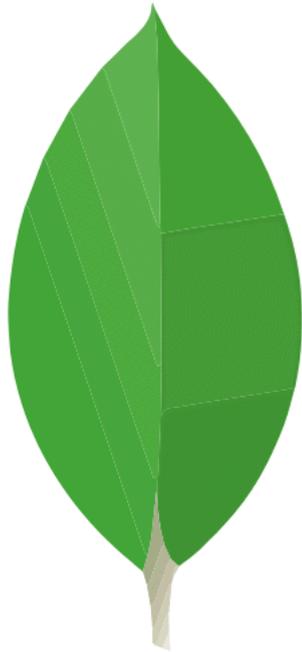
Applicazioni Correlate



-

Go Placeholder Image

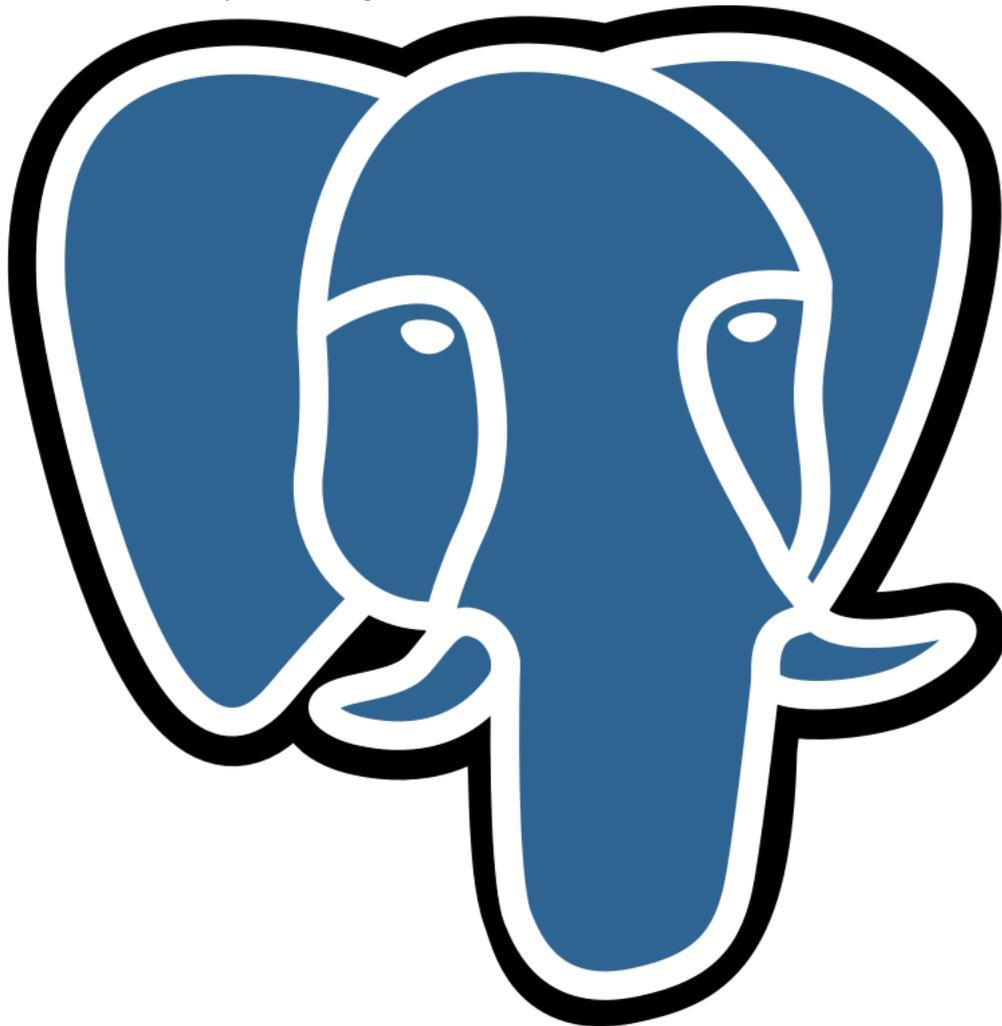
Applicazione in Go per la creazione di immagini segnaposto.
Docker Docker Compose Go JavaScript



•

Go MongoDB App

Applicazione basata su MongoDB ed implementata in Go con il driver ufficiale.
DockerDocker ComposeGoMongoDB



•

Go PostgreSQL App

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