

GABRIELE ROMANATO

Menu

Utilizzare Supabase con Go

Supabase è una piattaforma open-source che fornisce funzionalità simili a Firebase, ma basata su PostgreSQL. In questo articolo vedremo come interagire con Supabase utilizzando il linguaggio Go.

Configurazione di Supabase

Accedi alla dashboard di Supabase e crea un nuovo progetto. Una volta creato, vai nella sezione **API** per ottenere l'URL e la chiave anonima (anon key).

Installazione del client Go

Per interagire con Supabase in Go, possiamo utilizzare una libreria HTTP come `resty`. Installiamola con il comando:

```
go get github.com/go-resty/resty/v2
```

Autenticazione con Supabase

Per eseguire richieste autenticate, dobbiamo includere la chiave API nelle intestazioni delle richieste:

```
package main

import (
    "fmt"
    "github.com/go-resty/resty/v2"
)

const (
    supabaseURL = "https://your-project.supabase.co"
    supabaseKey = "your-anon-key"
)

func main() {
    client := resty.New()
    resp, err := client.R().
        SetHeader("apikey", supabaseKey).
        SetHeader("Authorization", "Bearer "+supabaseKey).
        Get(supabaseURL + "/rest/v1/your_table")

    if err != nil {
        fmt.Println("Errore nella richiesta:", err)
        return
    }

    fmt.Println("Risposta:", resp.String())
}
```

Inserire dati

Per aggiungere un nuovo record al database:

```
resp, err := client.R().
    SetHeader("apikey", supabaseKey).
    SetHeader("Authorization", "Bearer "+supabaseKey).
    SetHeader("Content-Type", "application/json").
    SetBody(`{"nome": "Mario", "cognome": "Rossi"}`).
    Post(supabaseURL + "/rest/v1/your_table")
```

Aggiornare dati

Per aggiornare un record esistente:

```
resp, err := client.R().
    SetHeader("apikey", supabaseKey).
    SetHeader("Authorization", "Bearer "+supabaseKey).
    SetHeader("Content-Type", "application/json").
    SetBody(`{"cognome": "Bianchi"}`).
    Patch(supabaseURL + "/rest/v1/your_table?id=eq.1")
```

Eliminare dati

Per eliminare un record:

```
resp, err := client.R().
    SetHeader("apikey", supabaseKey).
    SetHeader("Authorization", "Bearer "+supabaseKey).
    Delete(supabaseURL + "/rest/v1/your_table?id=eq.1")
```

Conclusione

In questo articolo abbiamo visto come interagire con Supabase utilizzando Go. Con pochi semplici passaggi è possibile connettersi a un database PostgreSQL, eseguire operazioni CRUD e gestire l'autenticazione.

Applicazioni Correlate

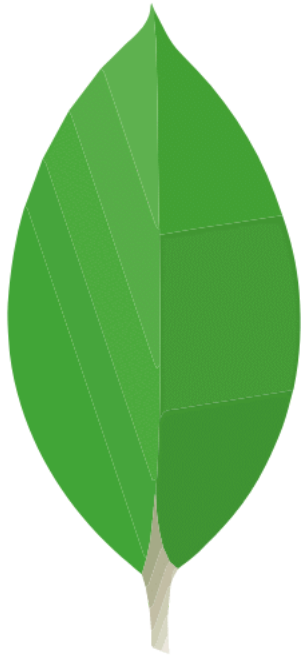


-

Go Placeholder Image

Applicazione in Go per la creazione di immagini segnaposto.

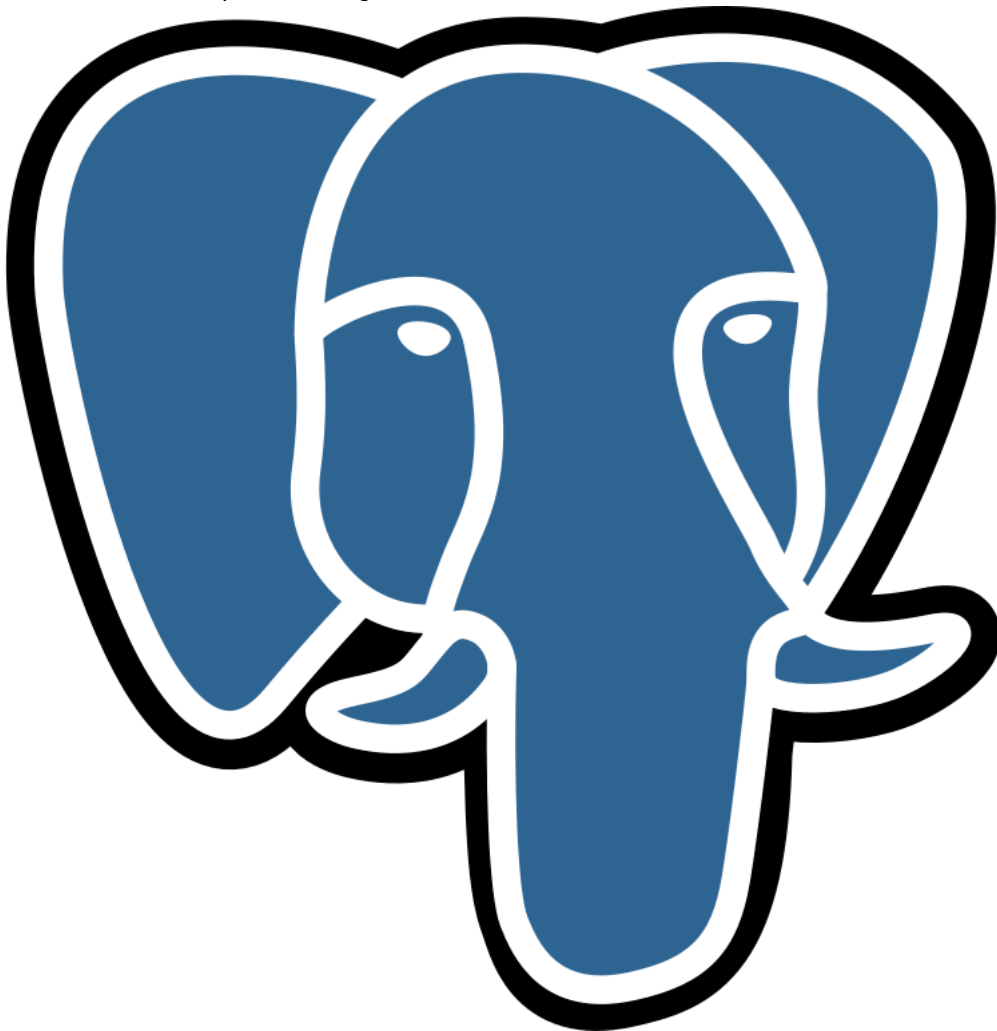
[Docker](#)[Docker Compose](#)[Go](#)[JavaScript](#)



•

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