

Coworking App -Backend- Middleware

Ivan <<**ossan**>> Pesenti
Software Developer



HACKERSSEN

POWERED BY

SORINT

MIDDLEWARES



HACKERSGEN

POWERED BY
SORINT

Definizione

Sono delle funzioni che **intercettano** (e possono modificare) le richieste e risposte HTTP che transitano dal nostro applicativo. Ogni middleware ha la facoltà di leggere/modificare la richiesta/risposta. Quest'ultima può essere intercettata da zero, uno o più middleware(s). L'ordine con cui vengono aggiunti conta: il primo aggiunto è il primo ad intercettare la richiesta e l'ultimo ad intercettare la risposta (sistema “**a pila**” o **LIFO**).



HACKERSGEN

POWERED BY

SORINT

Perchè usarli?

- ❑ Consentono di scrivere codice **modulare** e **mantenibile**
- ❑ Non dobbiamo **ripetere** della logica comune in tutti i nostri handlers
- ❑ Ogni handler si concentra sulla sua logica “core”
- ❑ Si rispetta il principio **SRP** (Single Responsibility Principle)
- ❑ Ci permette di applicare dei **filtri** sulle richieste e, quindi, fare arrivare ai nostri handlers solo quelle valide secondo le nostre logiche



HACKERSGEN

POWERED BY

SORINT

Esempi Concreti

- ❑ Middleware di autenticazione e autorizzazione
- ❑ Middleware di logging
- ❑ Middleware di gestione degli errori
- ❑ Middleware di validazione delle richieste
- ❑ Middleware di caching
- ❑ Middleware per il tracciamento delle richieste

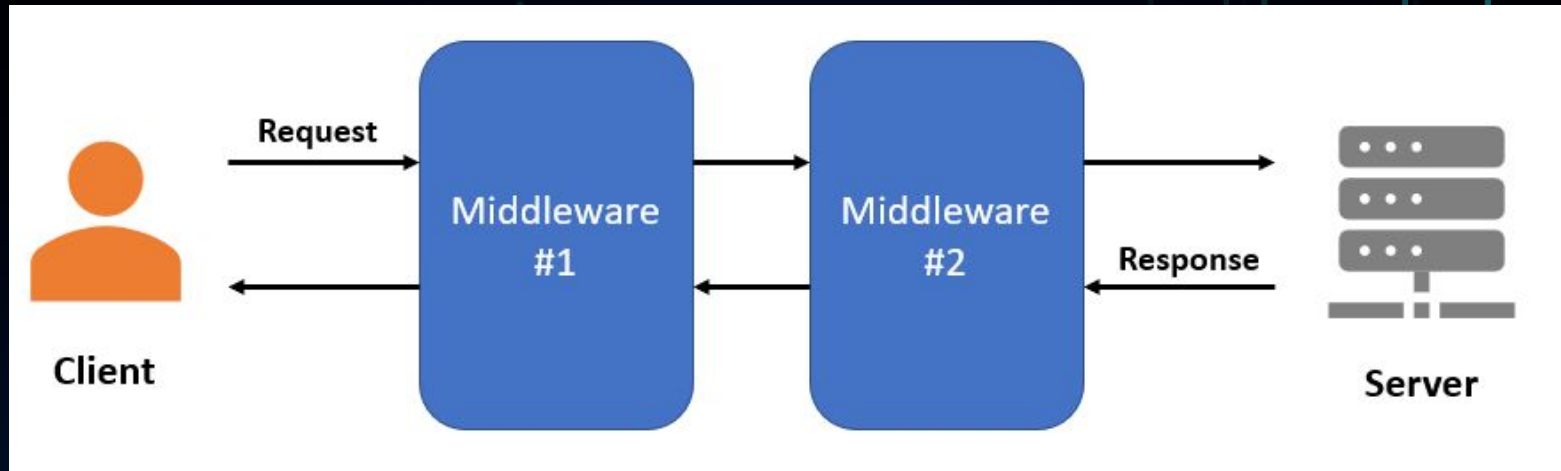


HACKERSGEN

POWERED BY

SORINT

Riassumendo...



HACKERSGEN

POWERED BY
SORINT

Steps per Implementazione in Go

1. Invocare il metodo **c.Use** su un'istanza di tipo gin.Engine
2. Il metodo c.Use si aspetta un'istanza del tipo **gin.HandlerFunc**
3. Il tipo gin.HandlerFunc è basato sul tipo **func(*gin.Context)**
4. Scrivere la logica del middleware
5. Runnare il metodo **c.Next** per far in modo che la richiesta arrivi al prossimo middleware (o all'handler)



HACKERSGEN

POWERED BY

SORINT

MIDDLEWARE IN GO



HACKERSGEN

POWERED BY
SORINT

THANKS!

@ossan



HACKERSGEN

POWERED BY

