package main

import (

"github.com/gin-gonic/gin"

"net/http"

)

// Global logging middleware

func loggingMiddleware() gin.HandlerFunc {

return func(c \*gin.Context) {

log.Println("Global logging middleware")

c.Next()

}

}

// Specific authentication middleware

func authMiddleware() gin.HandlerFunc {

return func(c \*gin.Context) {

if c.GetHeader("API-Key") != "secret" {

c.AbortWithStatusJSON(http.StatusUnauthorized, gin.H{"error": "unauthorized"})

return

}

c.Next()

}

}

func main() {

r := gin.Default()

// Apply logging middleware globally

r.Use(loggingMiddleware())

r.GET("/", func(c \*gin.Context) {

c.String(http.StatusOK, "Main handler")

})

// Apply authentication middleware specifically to the protected route

protected := r.Group("/protected")

protected.Use(authMiddleware())

protected.GET("/", func(c \*gin.Context) {

c.String(http.StatusOK, "Protected handler")

})

r.Run(":8080")

}