

ARTIGO: 11574

Golang - Check pagarme signature

```
import(
    "encoding/json"
    "encoding/hex"
    "crypto/sha1"
    "crypto/hmac"
    "strings"
    "fmt"
)

func (this *PagarmeController) Postback() {

    apiKey := "my api key"

    pagarmeSignature := this.Ctx.Request.Header.Get("X-Hub-Signature")

    if !strings.Contains(pagarmeSignature, "="){
        this.Abort("500")
        return
    }

    fmt.Println("*****")
    fmt.Println("** X-Hub-Signature = ", pagarmeSignature)
    fmt.Println("*****")

    cleanedSignature := strings.Split(pagarmeSignature, "=")[1]

    fmt.Println("*****")
    fmt.Println("** cleanedSignature = ", cleanedSignature)
    fmt.Println("*****")

    mac := hmac.New(sha1.New, []byte(apiKey))
    mac.Write(this.Ctx.Input.RequestBody)
    rawBodyMAC := mac.Sum(nil)
    computedHash := hex.EncodeToString(rawBodyMAC)

    fmt.Println("*****")
    fmt.Println("** computedHash = ", computedHash)
    fmt.Println("*****")

    if !hmac.Equal([]byte(cleanedSignature), []byte(computedHash)){
        fmt.Println("*****")
        fmt.Println("** Inválid Pagarme Signature: Expected: %v, Received: %v", string(cleanedSignature), string(computedHash))
        fmt.Println("*****")
        this.Abort("500")
        return
    } else {
        fmt.Println("*****")
        fmt.Println("** Válido Pagarme Signature: Expected: %v, Received: %v", string(cleanedSignature), string(computedHash))
        fmt.Println("*****")
    }

    this.Ctx.ResponseWriter.WriteHeader(200)
}
```