

Victor Nicéas & Lucas Alcantara

middleware - lista 5



Códigos importantes

Aplicação

Nossa aplicação consiste em fazer um **hashing** **SHA256** de uma mensagem.

```
type Response struct {  
    PwSha256 string  
}
```

```
//Request pro hasher  
type Request struct {  
    PwRaw string  
}
```

```
func hashRequest(req hashing.Request) string {  
    hashed := sha256.Sum256([]byte(req.PwRaw))  
    response := hex.EncodeToString(hashed[:])  
  
    return response  
}
```

Cliente



Cliente só chama o
método remoto

```
myReq := hashing.Request{PwRaw: message}  
  
response, err := hashing.HashPw(myReq,  
strings.ToUpper(transportProtocol))
```

Client Proxy



O nosso client proxy só
chama o requestor

```
response = requestor(req, transportProtocol)  
return response, err
```

Requestor



Responsável pela
**serialização e
coordenação** das
requisições

```
pwRawBytes, err := json.Marshal(req)
```

```
var response = Response{PwSha256: ""}
```

```
err = json.Unmarshal(CRH(pwRawBytes, transportProtocol),  
&response)
```

```
return response
```

CRH



Abre socket
Encerra conexão
Define timeouts

TCP

```
conn, err := net.DialTimeout(strings.ToLower(protocol),  
                             "localhost:3300",  
                             timeoutSeconds)  
  
_, err = conn.Write(pwRawBytes)  
  
response := make([]byte, 2048)  
n, err := conn.Read(response)  
return response[:n]
```

UDP

```
addr, err := net.ResolveUDPAddr(strings.ToLower(protocol),  
                                 "localhost:8030")  
  
conn, err := net.DialUDP("udp", nil, addr)  
  
_, err = conn.Write(pwRawBytes)  
  
response := make([]byte, 2048)  
n, err := conn.Read(response)  
return response[:n]
```

SRH



Handler recebe a requisição, repassa para o **invoker** e envia a resposta para o cliente

TCP

```
l, err := net.Listen("tcp", "localhost:3300")
conn, err := l.Accept()
receivedReq := make([]byte, 2048)
n, err := conn.Read(receivedReq)
conn.Write(invoker(receivedReq[:n]))
```

UDP

```
addr, err := net.ResolveUDPAddr("udp", ":8030")
conn, err := net.ListenUDP("udp", addr)
receivedReq := make([]byte, 2048)
n, addr, err := conn.ReadFromUDP(receivedReq)
conn.WriteToUDP(invoker(receivedReq[:n]), addr)
```


Invoker

Recebe requisição do **SRH**,
desserializa, invoca o
método remoto e envia a
resposta para o **SRH**

```
request := hashing.Request{PwRaw: ""}  
err := json.Unmarshal(receivedReq, &request)  
  
response := hashing.Response{PwSha256: hashRequest(request)}  
responseRaw, err := json.Marshal(response)  
  
return responseRaw
```