

# Reference Report

## Project 2 →

### 1. Client code

Source -

[https://github.com/tech-with-moss/go-usermgmt-grpc/blob/main/usermgmt\\_client/usermgmt\\_client.go](https://github.com/tech-with-moss/go-usermgmt-grpc/blob/main/usermgmt_client/usermgmt_client.go)

Which part of it do you reuse?

Line 18 - 25

Which part of your code reuse it?

Line 35-42 -> client/tokenclient.go

Why do you think it is necessary for you?

This code is needed as we need to set up a Client-Server so that client is able to connect to the server and then make requests. It helped me know the concept of gRPC server and various steps involved in creating one.

Also, this code was not copied as the following video gives the proper explanation of every single thing. So basically this video gave me a head start.

Link -> <https://www.youtube.com/watch?v=YudT0nHvkkE>

### 2. Server code

Source -

[https://github.com/tech-with-moss/go-usermgmt-grpc/blob/main/usermgmt\\_server/usermgmt\\_server.go](https://github.com/tech-with-moss/go-usermgmt-grpc/blob/main/usermgmt_server/usermgmt_server.go)

Which part of it do you reuse?

Line 28 - 36

Which part of your code reuse it?

Line 174-182 -> server/tokenserver.go

Why do you think it is necessary for you?

This code is needed as we need to set up a Client-Server so that client is able to connect to the server and then make requests. It helped me know the concepts of gRPC server and various steps involved in creating one.

Also, this code was not copied as the following video gives the proper explanation of every single thing. So basically this video gave me a head start.

Link - <https://www.youtube.com/watch?v=YudT0nHvkkE>

### 3. Storing tokens in Slice

The following video was used for reference

Link - <https://www.youtube.com/watch?v=QmldWTidEa8>

Also, I used the following link to determine which data structure would be better to persist the data and achieve concurrency.

Link - [https://go.dev/doc/faq#atomic\\_maps](https://go.dev/doc/faq#atomic_maps)

### 4. Drop Token Operation

Source-

<https://stackoverflow.com/questions/37334119/how-to-delete-an-element-from-a-slice-in-golang>

Which part of your code reuse it?

Line 63-64 -> server/tokenserver.go

Why do you think it is necessary for you?

As I was using slices to store all the tokens. It was necessary to understand how elements are removed from slices in golang. In the process I got this link which explains how it can be done. Hence to drop the token I used this source.

Approximate reused-code ratio:  $\frac{17}{300} = 0.056$

*Some other references which were used to inherit the concepts-*

<https://grpc.io/docs/languages/go/basics/>  
<https://stackoverflow.com/questions/44949467/when-do-you-embed-mutex-in-struct-in-go>  
<https://www.reply.com/alpha-reply/en/content/go-concurrency-with-mutex>  
<https://stackoverflow.com/questions/57700860/protoc-gen-go-program-not-found-or-is-not-executable>  
<https://zetcode.com/golang/module/>  
<https://gist.github.com/Integralist/48d35050d5342bb7568f04183b81ca29>  
<https://qvault.io/golang/golang-mutex/>  
<https://developers.google.com/protocol-buffers/docs/gotutorial>  
<https://gobyexample.com/command-line-flags>

Project 3 →

Most of the code was written from scratch as this was the extension of project 2. Below are some of the links which I referred to while working on this project. However, some of the code was reused from project 2 like setting up servers to serve client requests. Additionally, that concept was used to setup the RPC calls between two servers.

Approximate reused-code ratio:  $\frac{35}{550} = 0.063$

Links referred :-

<https://tutorialedge.net/golang/creating-simple-web-server-with-golang/>  
<https://stackoverflow.com/questions/63812531/golang-tcp-server-gives-dial-tcp-127-0-0-19999-connect-connection-refused-error>  
<https://stackoverflow.com/questions/14289256/cannot-convert-data-type-interface-to-type-string-need-type-assertion>  
<https://zetcode.com/golang/yaml/>  
<https://www.w3schools.io/file/yaml-arrays/>  
<https://www.geeksforgeeks.org/time-formatting-in-golang/>  
<https://stackoverflow.com/questions/3574716/date-and-time-type-for-use-with-protobuf>  
<https://pkg.go.dev/google.golang.org/protobuf/types/known/timestamppb>  
<https://stackoverflow.com/questions/25845172/parsing-rfc-3339-iso-8601-date-time-string-in-go>  
<https://medium.com/@greenraccoon23/multi-thread-for-loops-easily-and-safely-in-go-a2e915302f8b>  
<https://go.dev/blog/concurrency-timeouts>