**Тестовое задание для разработчика GoLang**

Реализовать клиент-серверное приложение на GoLang обеспечивающее обмен JSON сообщениями через вебсокет.

Приложение должно принимать на входе аргументы:  
clientserver [-c] <address> [port]  
Где:

ключ «c» - указание на запуск клиента

параметр <address> указывает сетевой интерфейс на котором запускается сервер или адрес подключения к серверу.

параметр port является не обязательным и указывает порт сервера. По умолчанию 7623.

Каждый запрос для каждого клиента имеет уникальный порядковый номер для клиента и сервера. Нумерация начинается с 1. Номер запроса используется для идентификации ответа на конкретный запрос.

Приложение в режиме сервера должно принимать входящие подключения. Для каждого нового подключения отправляется запрос информации о клиенте. До момента получения информации о клиенте никакие запросы от клиента не принимаются. Сервер периодически должен отправлять запрос ping.

Сервер выводит полученные сведения о клиенте и содержимое всех запросов от клиента и ответы клиенту в виде:  
*ID подключения, ID клиента, Запрос/Ответ: JSON запроса/ответа*

При получении запроса от клиента такой запрос должен быть переслан всем подключенным клиентам (Чат).

Клиент при подключении запрашивает синоним пользователя и подключается к серверу.

После подключения к серверу клиент может отправлять серверу запросы. Каждая введенная строка пользователя отправляется на сервер. Каждый полученный запрос типа «сообщение» от сервера должно отображаться на клиенте в виде:  
*ID отправителя: Текст сообщения*

**Типы запросов**

Информация о клиенте:  
{

id: 1,

function: info,

}

Ответ

{

id: 1,

response: info,

result: {

name: UserName

}

}

Пинг:

{

id: XX,  
 function: ping  
}  
Ответ  
{  
 id: XX  
 response: ping

result: {

timestamp: unixtimestamp (now)

}

}

Сообщение:

{

id: XX,

function: message,

data: «Text message»,

}  
Ответ  
{

id: XX,

response: message,

result: true|false

}