# Appendix B (Security of WebSocket connection)

## Why WebSocket?

For the synchronizing players in multiplayer game I need the message that goes from server with no request from client, therefore, base solution with REST can not be applied because only way of synchronizing with REST is “pinging” until both users do same.

## How it is protected

For every client send message, we check that header got “refresh” and “access” namespaces – they represent JWT with valid lifetime of 60 seconds and 1 hour respectively. The server at the middle layer checks if access token is valid – in success scenario it passes not changed message to server, otherwise it checks the refresh token – if it satisfy in message destination adding “/1” that points server that it should provide new tokens; if it also ended message not goes to the server and user receives errors that move him to log in page. In acting with web app can be the situation that user not the intruder but has not any tokens – in registering or logging for the first time in long time, in this case, configuration check if mentioned token “Credentials” that initialized verification not by tokens but by the destination of message – in another word, without tokens you can reach only register and login end points.

Basically, this kind of protection applied for the REST server communication.

## Why not in hand shake?

As was mentioned at the end of previous paragraph – not in all cases user got tokens, therefore, on connection event we can not know with what purposes user connecting to the server.

## Possible weak spots

For now, there is no verification of who subscribing on end points but there is no information that can be beneficial for stealer. In addition, for mostly parts user receive information that providing in the path that associated with his/her login, therefore, you should get someone login that has never mentioning outside of communication in the app. For now, I believe that time required to steal information to the significance of it is no worth.

## Other variants of protection

As a variant can be applied ticket technology – basically, same to JWT protection but the slightly different is token only one and being update every message. This is not being applied in project because the point was to study JWT technology, also, this variant requires little bit more resources of server and provide no benefits and more restricting ways how WebSocket can be used.

## Conclusion

The WebSocket still fresh technology and developer are not providing proper way of implementing security of the connection. My recommendation was not apply it to use in the communication with data that can be an object of attack.