Documento di rete gruppo AM50

Communication protocol

Communication between client and server is based on the exchange of JSON strings serialized and deserialized via GSON library.

Strings are sent via a PrintWriter through the socket output stream and are received via a BufferedReader on the socket input stream.

There are two main objects that are serialized to JSON respectively for messages from the client to the server and from the server to the client.

* Client to Server

Messages from the client to the server holds commands to be executed on the server so the JSON sent is structured as follows:

1. {
3. “command”: String,
4. “value1”: String,
5. “value2”: String
7. }

The command field holds a string mapped on the server with the relative method, value1 and value2 are optional fields used to pass parameters.

* Server to client

Messages from the server to the client are mostly responses or results of a command and are structured as follows:

1. {
2. “type”:String,
3. “context”:String,
4. “message”:String
5. }

The tipe field indicates the message type (error, confirmation, notify, islands dashboard ecc…), the context field is optional and used when managing errors or confirmations and contains the specific error code to be handled by the client, the message field can be used to send a simple message or store a nested JSON with more information, for example islands or dashboard state.

Communication flow

Communication between client and server begin whan a new client connects.

The server logs the connection and starts listening to messages from the client.

The first message sent by the client depends on user choise of starting a new game or joining an existing one.

* New Game

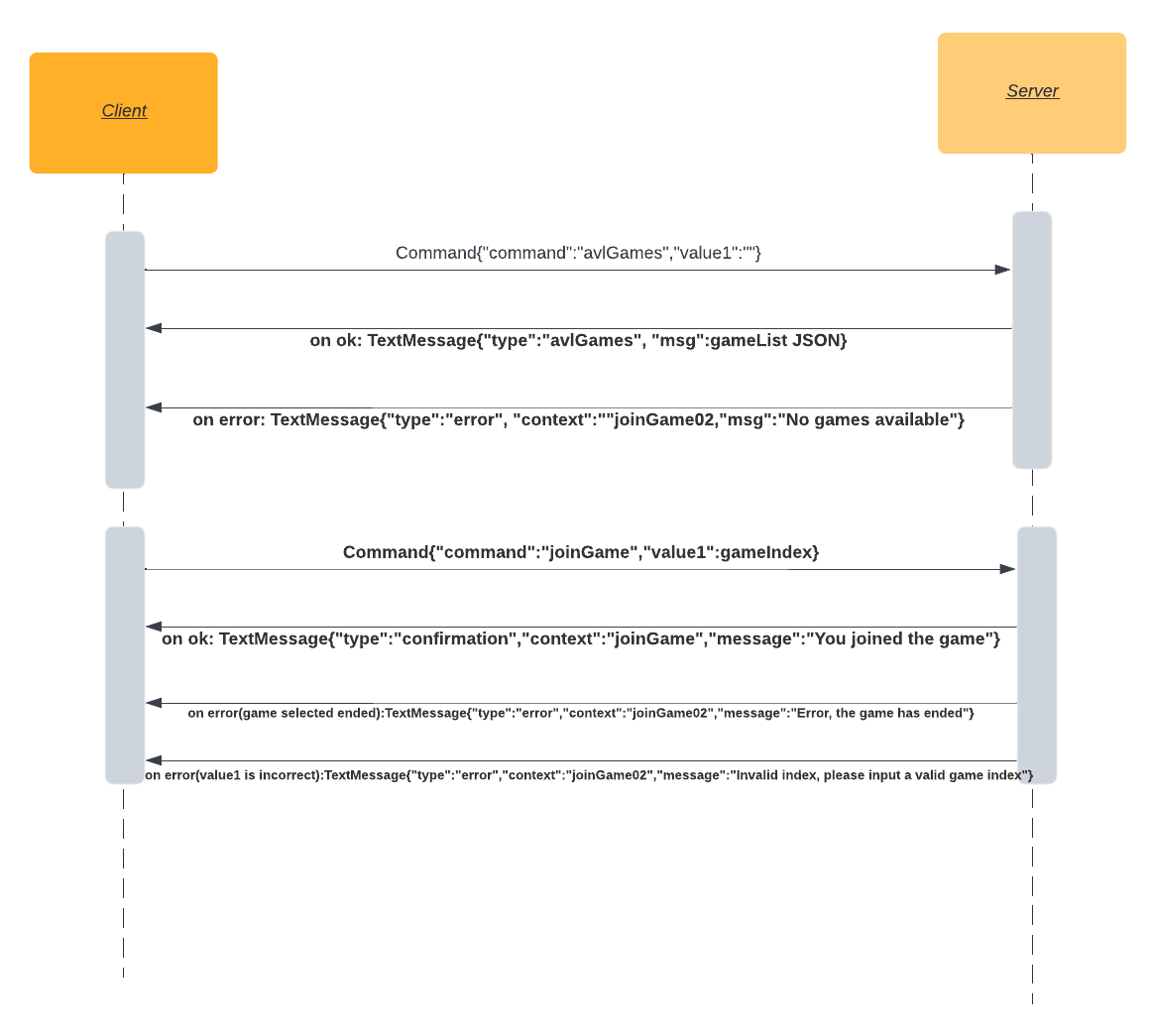
The user inputs the number of player desired, 2 or 3, a message is sent to the server.

Table

Description automatically generated with medium confidence

* Join Game

If the user chose to join a game the client asks the server for the list of available games and shows it to the user. Then sends a message to the server with the game index chosen by the user.



* Login

After starting a new game or to succefflully join a game the user needs to log in with a unic username.

