

# Protocols

The rules governing client-server interaction

Every transmission made by both client and server is sent as a **STRING** (so if the documentation below says that the server responds with 1, it means the string “1”), even the purely numerical transmissions

We note a couple important patterns:

1. The only requests clients can make of the server without having logged in a user are to either login a user or create a new account. All other requests for data must be preceded by the logging in of a user. If a request is made before a user has been logged in, the server returns -2.
2. A response of -1 from the server always means that there was a server error and something went wrong server-side

Now, we address each potential client-server interaction individually.

- Creating a new account for a user
  - Client – “create username password”
    - Note: the client should only accept usernames and passwords that do not contain either spaces or commas, otherwise corruption of data could occur. If the client does not check for this, the server will not allow the account to be created
  - Server – responds with one of the following:
    - -1 – server error; request could not be properly processed
    - 0 – account creation successful, client can now make requests on behalf the new user
    - 1 – the given username is already in use
    - 2 – the format of the username and/or password is incorrect
- Logging in a user
  - Client – “login username password”
  - Server – responds with one of the following:
    - -1 – server error; request could not be properly processed
    - 0 – login successful, client can now make requests on behalf of the given user
    - 1 – the given username does not exist
    - 2 – the given password is invalid for the given username
    - 3 – the command is invalid in some way
  - If the login is successful, the server then does the following:
    - First, it sends an integer, which corresponds to the number of ongoing games that the logged in user is involved in
    - Then it sends that many batches of transmissions, where each batch corresponds to the following data points, each sent on a separate line:
      - GameID
      - Name of White player
      - Name of Black player
      - State - 0 if it's White's turn, 1 if it's Black's
      - Turn – the current turn number
      - White check – 1 if White is in check, 0 otherwise
      - White checkmate – 1 if White is in checkmate, 0 otherwise
      - Black check -1 if Black is in check, 0 otherwise
      - Black checkmate – 1 if Black is in checkmate, 0 otherwise
    - All of these data points are sent as **STRINGS**

- If the server encounters a critical error, such as a flaw in its local database, it will send -1, but only BETWEEN batches, never in the middle of sending one. If part of a batch of data has been received, you may assume the transmission will proceed as normal at least until the end of the batch
  - The client, of course, is free to use as little or as much of this information as it pleases
- Creating a new game
  - If the client has already logged a user in, and the user wants to create a new game
  - Client – “creategame gameId”
    - Note: gameIDs cannot include commas, and must contains at least one alphabetical character
  - Server – responds with one of the following
    - -2 – the client does not have a user logged in, so the request couldn’t be handled
    - -1 – server error; request could not be properly processed
    - 0 – game was successfully created
      - Whichever user creates the game automatically plays White
    - 1 – the given game already exists
    - 2 – the given gameId is invalidly formatted
- Joining a game
  - If the client has already logged a user in, and the user wants to join a particular game
  - Client – “joingame gameId”
  - Server – responds with one of the following:
    - -2 – the client does not have a user logged in, so the request couldn’t be handled
    - -1 – server error; request could not be properly processed
    - 0 – the user has successfully joined the game
    - 1 – the given game does not exist
    - 2 – the given game already has two players
- Loading a game
  - If the client has already logged a user in, and the user wants to load a game that he is playing and view the board
  - Client – “loadgame gameId”
  - Server – responds with one of the following
    - -2 – the client does not have a user logged in, so the request couldn’t be handled
    - -1 – server error; request could not be properly processed
    - 0 – success; the user is a player in the given game, and the board data will be sent shortly
    - 1 – the given game does not exist
    - 2 – the user is not in the given game
  - If the server responded with success, it then sends the contents of the requested game’s data file, one line at a time. For details on how these files are formatted and how game data is encoded, see the file named Data in this directory
- Sending a move
  - If the client has already logged a user in, and the user wants to make a move in a game that he is playing
  - Client – “move gameId src\_row,src\_col->dest\_row,dest\_col”
    - src\_row,src\_col is the square occupied by the piece that is moving
    - dest\_row,dest\_col is the square the piece is moving to
    - Where row 0 is white’s back row, and column 0 is the first column from the left (from white’s perspective)
  - Server – responds with one of the following:

- -2 – the client does not have a user logged in, so the request couldn't be handled
  - -1 – server error; request could not be properly processed
  - 0 – move was successfully made; game has been updated
  - 1 – the move is invalid
  - 2 – no moves can be processed because a pawn needs to be promoted
  - 3 – it is not the user's turn to make a move
  - 4 – move was successfully made, but now the user needs to promote a pawn
- Logging out
  - If the user has logged out of the client, but the client doesn't want to close the connection yet, they should send a logout request so the server can free up some memory associated with the particular user that was signed in
  - This does not close the connection the client has with the server
  - Client – “logout”
  - The server does not respond, but the client should know that it can no longer make requests on behalf of the previously logged in user