**EACH VALUE (num\_players, username, etc) IS IMPLEMENTED IN ALL LOWERCASE e.g. {“type” : “play”}**

Gamestate info (lead player and backup player only)

|  |  |  |  |
| --- | --- | --- | --- |
| NUM\_PLAYERS | DEALER\_NAME | CUR\_TURN | TOTAL\_POT |
| PUBLIC\_CARDS | PLAYER\_INFO \* NUM\_PLAYERS | | |

Player state info (each player only)

|  |  |  |
| --- | --- | --- |
| USERNAME | NUM\_CHIPS | CARDS\_IN\_HAND |
| CUR\_BET |  |  |

Server player info (server only)

|  |  |  |
| --- | --- | --- |
| USERNAME | NUM\_CHIPS | LAST\_TABLE |
|  |  |  |

Server client socket info (server only)

|  |  |
| --- | --- |
| SOCKET\_NUMBER | DATA\_TO\_RECEIVE |

Server table info (server only)

|  |  |  |
| --- | --- | --- |
| NUM\_PLAYERS | HOST | PORT |

Client Request => Server

|  |  |
| --- | --- |
| TYPE (“cash” or “play” or “buy”) | USERNAME |

Process is

1. Client Sends:

|  |
| --- |
| DATA\_SIZE\_TO\_SEND |

1. Sever Sends ACK:

|  |
| --- |
| DATA\_SIZE\_TO\_RECEIVE (data\_size\_to\_send back to client) |

1. Client Sends Request Data to Server

Server response => client req (play)

|  |  |  |
| --- | --- | --- |
| HOST | PORT | NEW\_TABLE (“True” or “False”) |

Process is

1. Server Sends:

|  |
| --- |
| DATA\_SIZE\_TO\_SEND |

1. Client Sends ACK:

|  |
| --- |
| DATA\_SIZE\_TO\_RECEIVE (data\_size\_to\_send back to client) |

1. Server Sends Request Data to Client

Peer connection => Main Peer

Process is

1. Peer connects to socket
2. Peer sends:

|  |  |
| --- | --- |
| USERNAME | NUM\_CHIPS |