

Communication is accomplished through JSON payloads sent via a WebSocket. Messages should be in the format,

```
{ "type" : "join", "name" : "Jeff" }
```

Client to Server

type	data
join	String name
leave	-
updateSettings	String name
playSubmission (when a player submits a card)	int[] cardIDs
submissionsJudged (when the judge submits their decision)	int winningPlayerID (sourced from Response[].submitter as seen in the judgeSubmissions type)
submissionRead (when the judge reads a submission)	int[] cardIDs
nextRound	-

Server to Client

type	data
didQueue	-
didJoin	int playerID
judging (when the judging period starts)	-
gameSync	Player player, int judge (the player's Player object contains their updated hand)
judgeSubmissions (tell the judge to make his/her	Response[] responses (each response has:

decision)	Card[] cards, int submitter)
roundStarted	String prompt, int numOfBlanks
roundEnded	-
response	int code (0 or error code)

error code	meaning
-1	Client sent a message with an invalid type.
1	Player tried to submit the wrong number of cards (i.e. of of cards submitted doesn't match black card's numBlanks).
2	Judge submitted an invalid winningPlayerID (their own ID or out of range).
3	Judge tried to judge before all cards were read.
4	Client tried to join with a blank name.
5	Client tried to start a round when a round was already in progress.
6	Client tried to start a round without sufficient players.