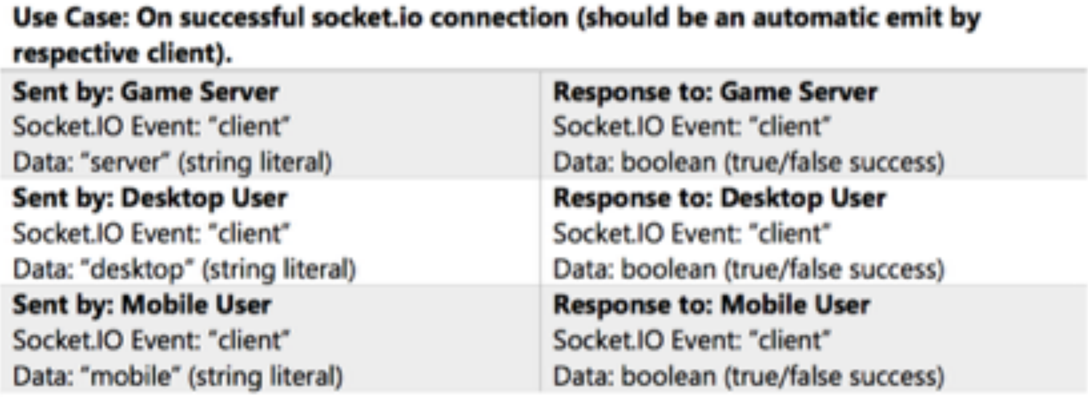
ANEXD

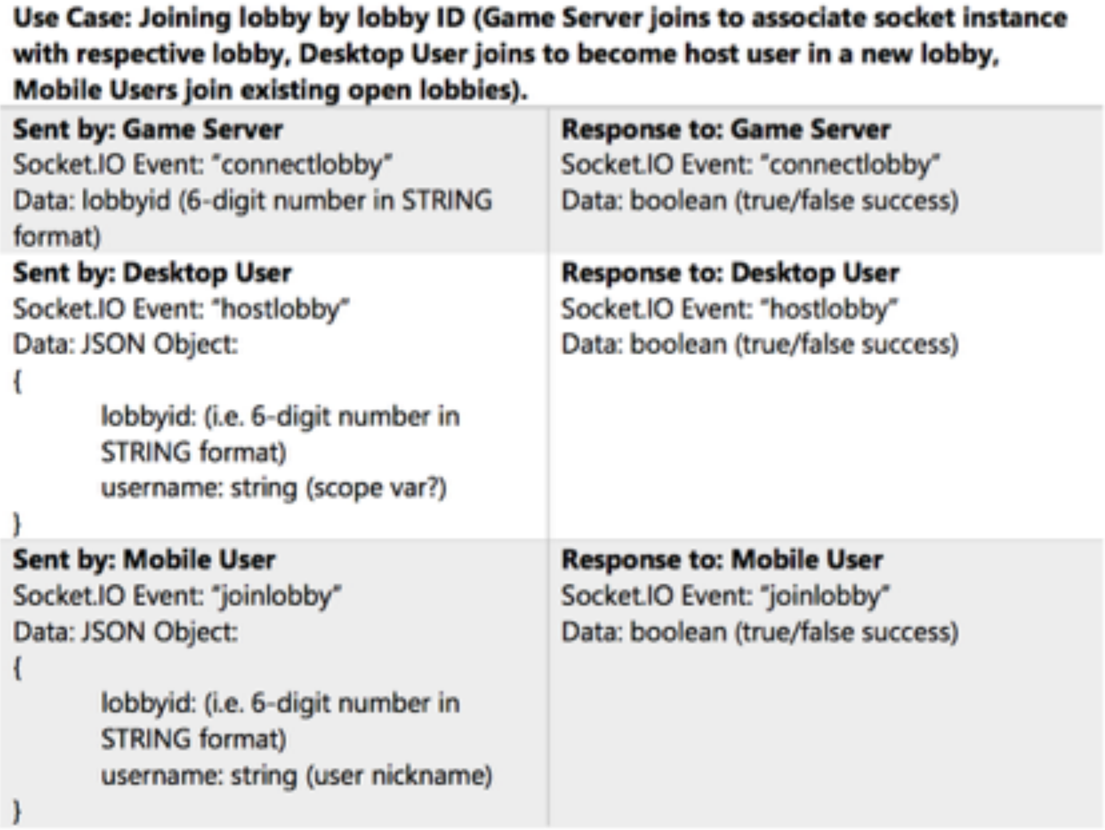
Frederick Harrington
Fred Barnes
CO600: SocketIO Messages
16 October 2015

Socket.io Message standards



Use Case: On successful socket.io connection (should be an automatic emit by respective client).

Sent by: Game Server Socket.IO Event: "client" Data: "server" (string literal)	**Response to: Game Server** Socket.IO Event: "client" Data: boolean (true/false success)
Sent by: Desktop User Socket.IO Event: "client" Data: "desktop" (string literal)	**Response to: Desktop User** Socket.IO Event: "client" Data: boolean (true/false success)
Sent by: Mobile User Socket.IO Event: "client" Data: "mobile" (string literal)	**Response to: Mobile User** Socket.IO Event: "client" Data: boolean (true/false success)

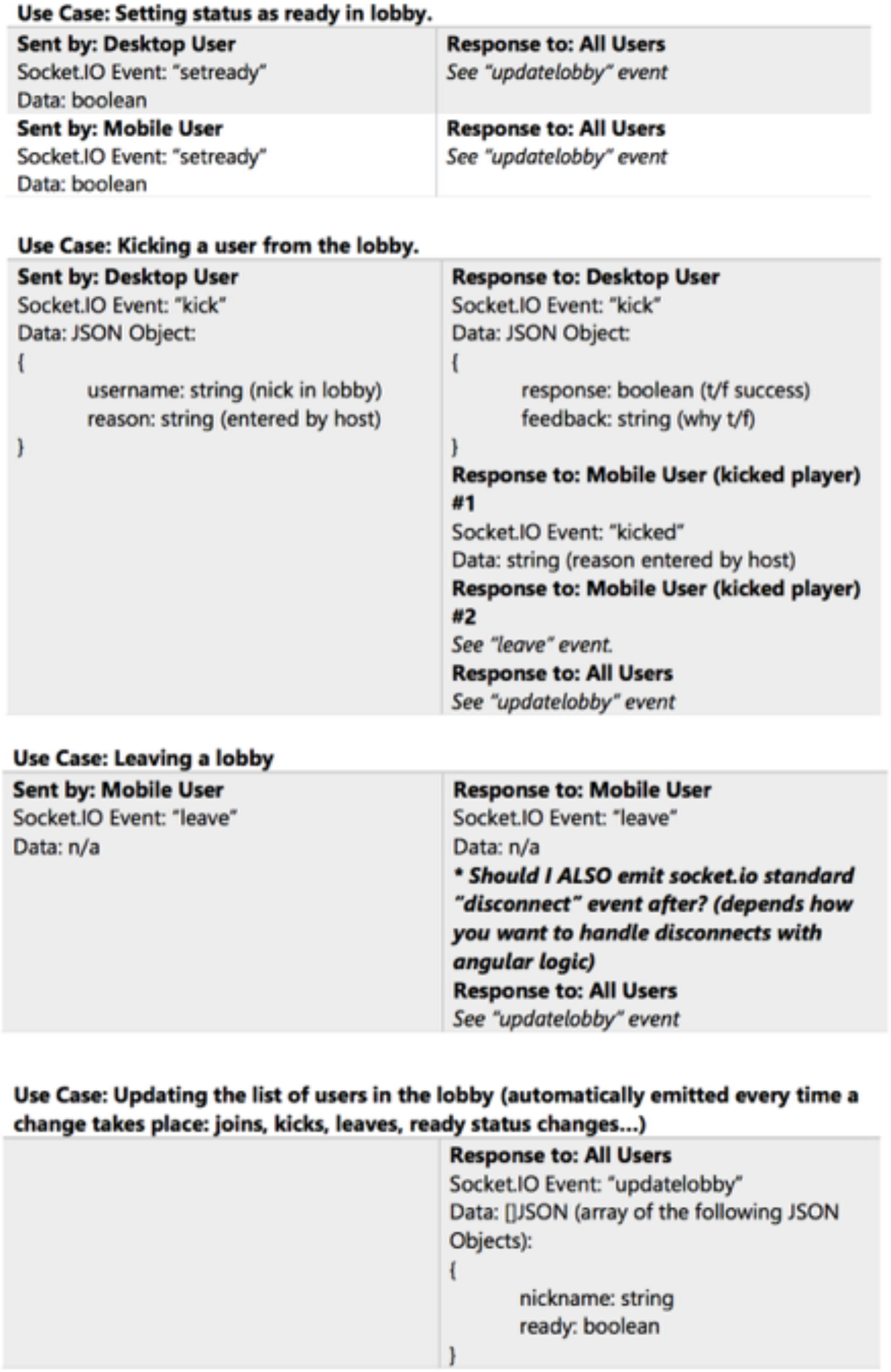


Use Case: Joining lobby by lobby ID (Game Server joins to associate socket instance with respective lobby, Desktop User joins to become host user in a new lobby, Mobile Users join existing open lobbies).

Sent by: Game Server Socket.IO Event: "connectlobby" Data: lobbyid (6-digit number in STRING format)	**Response to: Game Server** Socket.IO Event: "connectlobby" Data: boolean (true/false success)

Sent by: Desktop User
Socket.IO Event: "hostlobby"
Data: JSON Object:
{
lobbyid: (i.e. 6-digit number in STRING format)
username: string (scope var?)
}
	Response to: Desktop User
	Socket.IO Event: "hostlobby"
	Data: boolean (true/false success)

Sent by: Mobile User
Socket.IO Event: "joinlobby"
Data: JSON Object:
{
lobbyid: (i.e. 6-digit number in STRING format)
username: string (user nickname)
}
	Response to: Mobile User
	Socket.IO Event: "joinlobby"
	Data: boolean (true/false success)



Use Case: Setting status as ready in lobby.

Sent by: Desktop User Socket.IO Event: "setready" Data: boolean	**Response to: All Users** *See "updatelobby" event*
Sent by: Mobile User Socket.IO Event: "setready" Data: boolean	**Response to: All Users** *See "updatelobby" event*

Use Case: Kicking a user from the lobby.

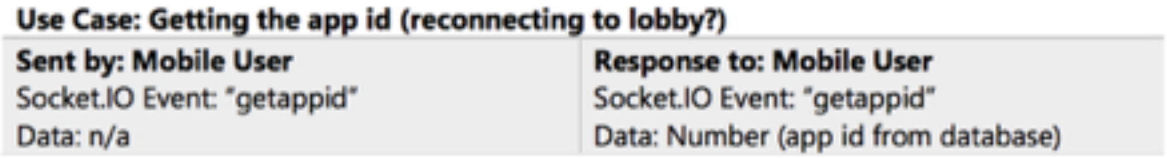
Sent by: Desktop User
Socket.IO Event: "kick"
Data: JSON Object:
{
username: string (nick in lobby)
reason: string (entered by host)
}
	Response to: Desktop User
	Socket.IO Event: "kick"
	Data: JSON Object:
	{
	response: boolean (t/f success)
	feedback: string (why t/f)
	}
	Response to: Mobile User (kicked player) #1
	Socket.IO Event: "kicked"
	Data: string (reason entered by host)
	Response to: Mobile User (kicked player) #2
	See "leave" event.
	Response to: All Users
	See "updatelobby" event

Use Case: Leaving a lobby

Sent by: Mobile User Socket.IO Event: "leave" Data: n/a	**Response to: Mobile User** Socket.IO Event: "leave" Data: n/a **** Should I ALSO emit socket.io standard "disconnect" event after? (depends how you want to handle disconnects with angular logic)*** **Response to: All Users** *See "updatelobby" event*

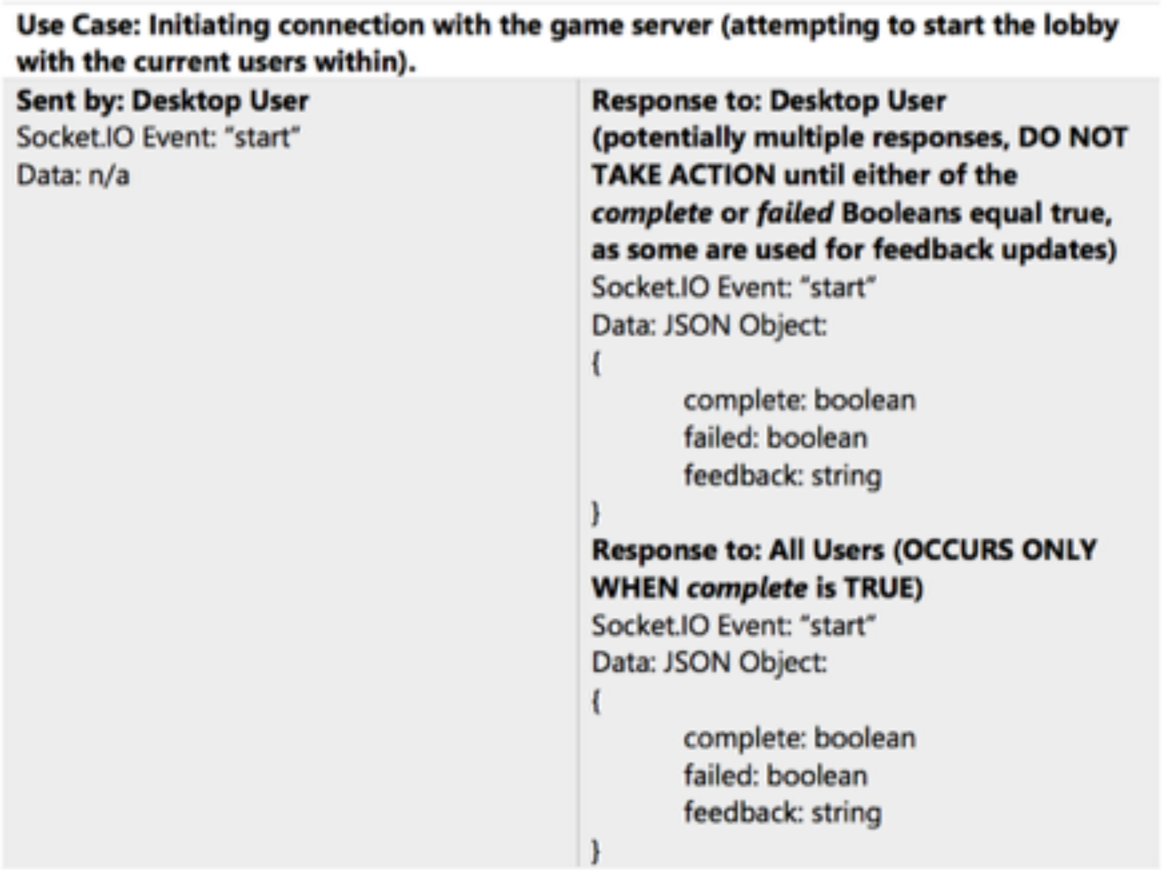
Use Case: Updating the list of users in the lobby (automatically emitted every time a change takes place: joins, kicks, leaves, ready status changes…)

	Response to: All Users
	Socket.IO Event: "updatelobby"
	Data: []JSON (array of the following JSON Objects):
	{
	nickname: string
	ready: boolean
	}



Use Case: Getting the app id (reconnecting to lobby?)

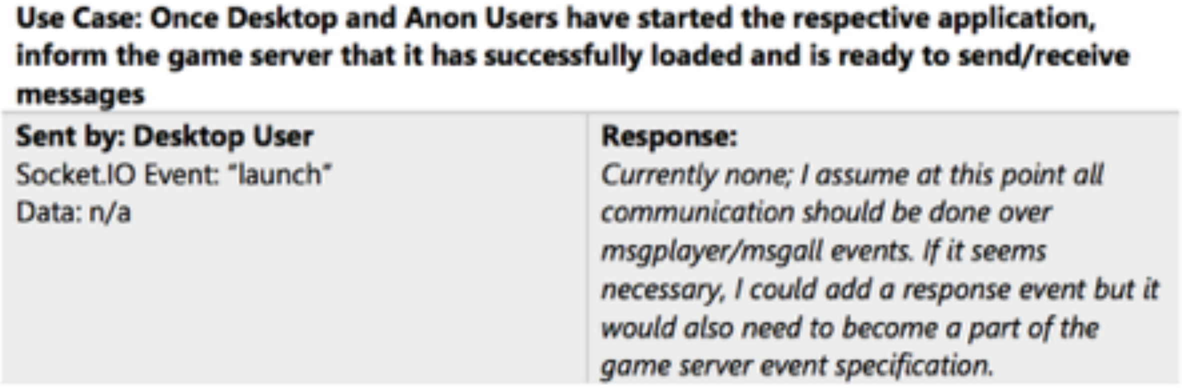
Sent by: Mobile User	**Response to: Mobile User**
Socket.IO Event: "getappid" Data: n/a	Socket.IO Event: "getappid" Data: Number (app id from database)



Use Case: Initiating connection with the game server (attempting to start the lobby with the current users within).

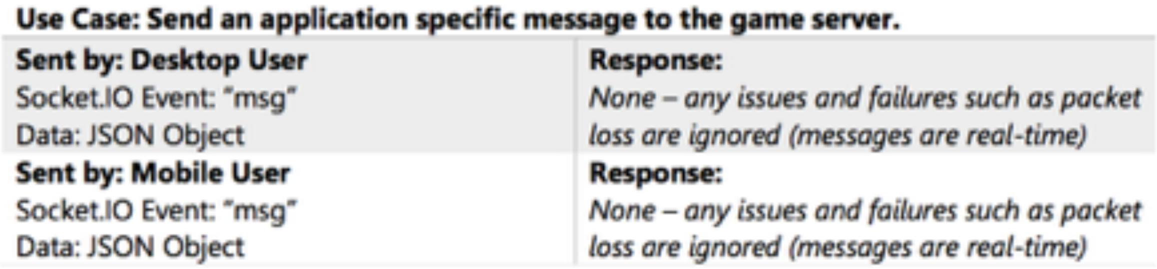
Sent by: Desktop User	**Response to: Desktop User (potentially multiple responses, DO NOT TAKE ACTION until either of the *complete* or *failed* Booleans equal true, as some are used for feedback updates)**

Socket.IO Event: "start"
Data: n/a
	Socket.IO Event: "start"
	Data: JSON Object:
	{
	complete: boolean
	failed: boolean
	feedback: string
	}
	Response to: All Users (OCCURS ONLY WHEN *complete* is TRUE)
	Socket.IO Event: "start"
	Data: JSON Object:
	{
	complete: boolean
	failed: boolean
	feedback: string
	}



Use Case: Once Desktop and Anon Users have started the respective application, inform the game server that it has successfully loaded and is ready to send/receive messages

Sent by: Desktop User	**Response:**
Socket.IO Event: "launch" Data: n/a	*Currently none; I assume at this point all communication should be done over msgplayer/msgall events. If it seems necessary, I could add a response event but it would also need to become a part of the game server event specification.*



Use Case: Send an application specific message to the game server.

Sent by: Desktop User Socket.IO Event: "msg" Data: JSON Object	**Response:** *None – any issues and failures such as packet loss are ignored (messages are real-time)*
Sent by: Mobile User Socket.IO Event: "msg" Data: JSON Object	**Response:** *None – any issues and failures such as packet loss are ignored (messages are real-time)*