

### Milestone 3 API definitions

#### Game: **Texas Hold 'Em**

Action	Input/Data	Pre condition(s)	Post condition(s)	API Endpoint
<b>User creates an account</b>	1. Username 2. Password	1. Username is not already being used	1. An account gets created with the user's information 2. User gets logged in	<b>POST</b> /account/create
<b>User logs in</b>	1. Username 2. Password	1. Username exists 2. Password is correct	1. User gets logged in 2. User gets a session token	<b>POST</b> /account/login
<b>User logs out</b>	1. user_id		1. User gets logged out	<b>POST</b> /account/logout
<b>User creates a game</b>	1. user_id		1. A new game is created (creating a game_id) 2. user_id is added to game_id 3. Player is redirected to the game room 4. user_id is added as the host of this game	<b>POST</b> /games/create
<b>User joins a game</b>	1. user_id 2. game_id	1. Game exists	1. player_id is added to game_id 2. Player is redirected to the game room	<b>POST</b> /games/{game_id}
<b>User creates a chat with another player</b>	1. fromUser_id 2. toUser_id	1. Both users exist	1. Creates a new conversation, with a convo_id	<b>POST</b> /chats
<b>User sends a chat message</b>	1. fromUser_id 2. convo_id 3. message		1. message gets sent to the chat between both users	<b>POST</b> /chats/{convo_id}/messages

<b>User gets a conversation</b>	1. user_id 2. convo_id		1. The conversation between the two users gets sent	<b>GET</b> /chats/{convo_id}/messages
<b>User sends an in-game chat message</b>	1. fromUser_id 2. game_id 3. message	1. fromUser_id is inside of the game specified by game_id	1. message gets sent to the in-game chat chat	<b>POST</b> /games/{game_id}/messages
<b>User gets the in-game chat</b>	1. user_id 2. game_id		1. In-game chat gets updated to reflect the current status of the chat	<b>GET</b> /games/{game_id}/messages
<b>Get game state</b>	1. game_id		1. Get the game's information including player details, community cards, current bet amounts, and game history	<b>GET</b> /games/{game_id}
<b>Host starts a game</b>	1. user_id 2. game_id	1. user_id is the id of the person who created the game 2. Game has more than 1 player	1. Round will start 2. Game UI for each player will activate	<b>POST</b> /games/{game_id}/start
<b>Start Round/Deal Hole cards</b>	1. game_id	1. There is not a current round in progress	1. Deal 2 cards to each player 2. Set the community cards 3. Round start gets displayed in the game history	<b>POST</b> /games/{game_id}/startRound
<b>Player calls</b>	1. game_id 2. user_id	1. It is the user's turn 2. There is something in the pot	1. Player's bid raises to the current blind 2. It becomes the next player's turn 3. Player's action gets displayed in the game history	<b>POST</b> /games/{game_id}/players/{user_id}/call
<b>Player raises bet</b>	1. game_id 2. user_id	1. It is the user's turn	1. Pot gets increased	<b>POST</b> /games/{game_id}/

	3. betAmount		2. Player's bid gets raised 3. Player gets marked as the "raiser" 4. It becomes the next player's turn 5. Player's action gets displayed in the game history	players/{user_id}/bet
<b>Player folds</b>	1. game_id 2. user_id	1. It is the user's turn	1. Player's hand gets discarded 2. Player loses any money they put into the pot 3. It becomes the next player's turn 4. Player's action gets displayed in the game history	<b>POST</b> /games/{game_id}/players/{user_id}/fold
<b>Player checks</b>	1. game_id 2. user_id	1. It is the user's turn 2. There is nothing in the pot	1. It becomes the next player's turn 2. Player's action gets displayed in the game history	<b>POST</b> /games/{game_id}/players/{user_id}/check
<b>Deal next community card</b>	1. game_id	1. There has been a full loop starting from the "raiser" 2. More than 1 player is still in	1. The next community card gets revealed 2. Comm card getting dealt gets displayed in the game history	<b>POST</b> /games/{game_id}/dealCommCard
<b>End Round</b>	1. game_id	1. All 5 community cards are showing and there has been a full loop starting from the "raiser" 2. Only 1 player is still in (Everyone else folded)	1. Determine who won the round 2. Increase or decrease each player's purse depending on the outcome 3. Start next round if more than 1 person still has money 4. Winner of the round gets displayed in the	<b>POST</b> /games/{game_id}/endRound

# Team J

			game's history	
<b>End Game</b>	1. game_id	1. There is no round in progress 2. There is only 1 player with money remaining	1. The player with money left is declared the winner 2. Everyone gets sent back to the home page 3. Game gets deleted off of the server	<b>DELETE</b> /games/{game_id}