

Action	Inputs/Data	Pre Condition(s)	Post Condition(s)	API Endpoint
User registers an account	username, email, password	Username does not exist already in the database  Email does not exist already in the database		POST /user/register  { username, email, password }
User logs in	username, password	User has registered an account		POST /user/login  { username, password }
User logs out	user	User must be logged in	User session is destroyed	GET /user/logout  (gets user from sessions)
User creates a game	room (title), players (required), userID	User must be logged in, A game with the same room title cannot exist already	A new game is created in games table	POST /api/game/create  { room, players } (gets userID from session)
User joins a game	gameID, userID	User must be logged in	A new row is created in the game_players table	POST /api/game/join/:id  (gets userID from session, gameID is provided in the url)

User starts a game	gameID, userID	User must be logged in, User must be the owner of the game	The deck is created with a hand of 7 card for each player (user_id = userID), The draw deck is created (user_id = 0), and one card is put into play (user_id = -2)	POST /api/game/start/:id  (gets userID from sessions, gameID is provided in the url)
User plays a card	gameID, userID, cardID (canonical_cards.id),	User must be a player in the game, It must be the users turn, The card must be same color, same value, or a wild or +4 card	The card that the player plays gets updated to be the card in play (user_id = -2), The previous card in play gets discarded (user_id = -1)	POST /api/game/move/:id  { cardID }  (gets userID from sessions, gameID is provided in the url)
User draws a card	gameID, userID	User must be a player in the game, It must be the users turn	The card that the player draws either gets put into the players hand (user_id = userID) or if it is a valid move then the card must be played (user_id = -2), If the card is played then the previopus card in play gets discarded (user_id = -1)	POST /api/game/draw/:id  { cardID }  (gets userID from sessions, gameID is provided in the url)