

# Game API Design

Ghaida Sulaiman, Veronica Yuen, Daejuan Graham, Jenevieve Mendoza  
Group z-i

# How do you play UNO?

Goal:

Be the first to reach 500 points. Earn points by being the first player to discard all the cards in your hand.

Discarding cards

To discard cards, a player must match the number, color, or action of the card that is on the top discard pile. Unless the player has a wild card

# List of endpoints

Create game	POST /api/games/:game_id/create
Join game	POST /api/games/:game_id/join
Start game	POST /api/games/:game_id/start
Draw card	POST /api/games/:game_id/draw
Play card	POST /api/games/:game_id/card/:card_id/play
End round	GET /api/games/:game_id/end
Select wild color	POST /api/games/:game_id/color/:color
Send message	POST /api/games/:game_id/chat

# Deep dive - Play Card

Method and route: POST

/api/games/:game\_id/card/:card\_id/play

Purpose: A player plays a card from their hand

Authorization:

- User must be authenticated
- User must be in the game
- Must be the user's turn
- Game has to be in an active state

Validation checks:

- Game with game\_id exists
- User is a player in the game
- User who is trying to play a card is the active player
- The card can be played
- The card exists in the player's hand

State updates:

- Update player's had
- Update top discard card
- Advance turn
- Insert into game log

Success response: 202 Accepted

Error cases:

- 401 Unauthorized - not logged in
- 403 Forbidden - not current player
- 400 Bad request - invalid move
- 404 Not found - Game/card doesn't exist
- 409 Conflict - Game state changed

Socket.io Events

- game:state:update → all players in game
- game:hand:update → Player who played the card
- game:hand:draw → Player who has to draw

# Deep dive - Draw Card

Method and route: POST

/api/games/:game\_id/draw

Purpose: A player chooses to draw a card

Authorization:

- User must be authenticated
- User must be in the game
- Must be the user's turn
- Game has to be in an active state

Validation checks:

- Game with game\_id exists
- User is a player in the game
- User who is trying to play a card is the active player

State updates:

- Update player's hand
- Advance turn
- Insert into game log

Success response: 202 Accepted

Error cases:

- 401 Unauthorized - not logged in
- 403 Forbidden - not current player
- 404 Not found - Game doesn't exist
- 409 Conflict - Game state changed

Socket.io Events

- game:state:update → all players in game
- game:hand:update → Player who played the card

## Public events

- game:start
- game:player:joined
- game:state:update
- game:state:ended
- game:chat:updated
- lobby:chat:updated
- lobby:game:created
- Lobby:game:updated

## Private events

- game:hand:draw
- game:hand:update
- game:error

- game:state:reversed
- game:state:skipped