

CSC 667 - Team H

Allawzi Izzaldin, Ansh Khandelwal, Jose Pascua, Ajmal Wahab

Uno Game API

- 2-4 player turn-based UNO game
- The server validates all actions
- The API controls rooms, players, turns, and actions
- Real time sync via Socket.io

Registration and User Endpoints

- Auth:
 - POST /api/auth/register - Create user
 - POST /api/auth/login - Authenticate
 - POST /api/auth/logout - End session
- User:
 - GET /api/users/me - Current user
 - PATCH /api/users/me - Update username

Lobby and Room Management Endpoints

- POST /api/rooms - Create room
- GET /api/rooms?status=waiting|in_progress - List rooms
- POST /api/rooms/:roomId/join - Join room
- POST /api/rooms/:roomId/leave - Leave room
- POST /api/rooms/:roomId/start - Start game
- POST /api/rooms/:roomId/state - Get state

Game Logic Endpoints

- POST /api/games/:roomId/play-card
- POST /api/games/:roomId/draw
- POST /api/games/:roomId/pass
- POST /api/games/:roomId/call-uno

Example: play-card

- Validates turn, card owner, and if it is legal
- Updates the discard pile and next player

Message and Real-Time Updates

- HTTP:
 - POST /api/rooms/:roomId/chat/messages
 - GET /api/rooms/:roomId/chat/messages
- [Socket.io](#) Events:
 - game:state:update
 - game:player:joined
 - game:hand:update

Deep Dive - Start Game Endpoint

- Route: POST api/rooms/:roomID/start
- Validation Checks:
 - User is authenticated, User is the room creator, rooms status is waiting, and 2 -4 players are in room
- Edge Cases:
 - Not enough players
 - Players leave right before the game starts
 - Duplicate requests (Start being called twice)
- State Updates:
 - Deal 7 cards to each player
 - Initialize starting player and discard pile
 - Set rooms status to in_progress
- [Socket.io](#) Events triggered:
 - game:state:update (for all players) & game:hand:update (for each players hand)

Deep Dive - Play Card Endpoint

- Route: POST api/rooms/:roomID/play-card
- Validation Checks:
 - Player is authenticated and in room, Game is in progress, It is the players turn, player owns the card, and card is legal to be played.
- Edge Cases:
 - Players tried to play an illegal card
 - Special cards stacking or combos
 - Player wins on current play (last card)
- State Updates:
 - Remove card from players hand and add card to discard pile
 - Apply effects
 - Advance turn / check for winner
- [Socket.io](#) Events triggered:
 - Game:state:update(all players) , game:hand:update (current player) , game:ended (if a player wins)

Socket.io Events Summary

- Public
 - game:state:update - Turn updates
 - game:player:joined - New player joined
 - game:ended - Winner and final result
 - chat:message - Speak in chat
- Private:
 - Game:hand:update - Players updated hand