System Operations

Name: view lobbies

Description: view all currently existing game lobbies that the user can join

Operation Type: QUERY

Exposition: EXTERNAL

Roles: User

Related FR: 2.1.

Pre-conditions:

+user is authorised

Post-conditions:

+all joinable lobbies are returned

Request: +maxShowAmount

Response: +lobbies[]

GET /lobbies/

Request:

{

"maxShowAmount": int

}

Response:

{

"lobbies":

{

{"lobbyId": int, "hostRating": int, "format": {...}},

{"lobbyId": int, "hostRating": int, "format": {...}},

…

{"lobbyId": int, "hostRating": int, "format": {...}}

}

}

Name: join lobby

Description: join a currently existing game lobby that the user can join

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: User

Related FR: 2.2.

Pre-conditions:

+user is authorised

+the lobby exists

+user has a rating within reasonable bounds of the lobby creator's

+user is not currently in a game

+user is not currently hosting a lobby

+lobby host is not currently in a game

Post-conditions:   
 +game with the lobby host of the appropriate lobby format is started

+lobby is destroyed

Request:

+lobbyId

Response:

+gameId

POST /lobbies/{id}

Request:

{

"lobbyId": int

}

Response:

{

"lobbies":

{

{"lobbyId": int, "hostRating": int, "format": {...}},

{"lobbyId": int, "hostRating": int, "format": {...}},

…

{"lobbyId": int, "hostRating": int, "format": {...}}

}

}

Name: view lobbies of a certain format

Description: view all currently existing game lobbies of a certain format that the user can join

Operation Type: QUERY

Exposition: EXTERNAL

Roles: User

Related FR: 2.3.

Pre-conditions:

+user is authorised

+the format is valid

Post-conditions: +all joinable lobbies of specified format are returned

Request:

+maxShowAmount

+affiliationDistribution (black|white|random)

+timeTotal

+timeIncrement

+ranked

Response: +lobbies[]

GET /lobbies/

Name: create lobby

Description: create a game lobby that another user can join

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: User

Related FR: 2.4.

Pre-conditions:

+user is authorised

+user is not currently hosting a lobby

+user is not currently in a game

Post-conditions:

+a lobby is created and the user joins it

Request:

+affiliationDistribution (black|white|random)

+timeTotal

+timeIncrement

+ranked

Response: +lobbyId

POST /lobbies/new

Name: view ongoing game state

Description: get all information about the game state of the chess board of a game

Operation Type: QUERY

Exposition: EXTERNAL

Roles: User

Related FR: 2.5.1.

Pre-conditions:

+user is authorised

+the game exists

+user is one of the players in the game

Post-conditions:

+the full game state is returned

Request:

+gameId

Response:

+gameState

GET /games/{id}

Name: make a move

Description: make a move in the currently ongoing game

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: User

Related FR: 2.5.2.

Pre-conditions:

+user is authorised

+the game exists

+user is one of the players in the game

+it is the user's turn

+the move is valid

Post-conditions:

+the move is made, game state is updated

+it is the other player's turn

Request:

+gameId

+Move

Response:

+gameState

PUT /games/{id}/move

Name: forfeit

Description: forfeit the currently ongoing game

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: User

Related FR: 2.5.3.

Pre-conditions:

+user is authorised

+the game exists

+user is one of the players in the game

Post-conditions:

+the player loses, game state is updated

Request:

+gameId

Response:

+gameState

POST /games/{id}/forfeit

Name: offer a draw

Description: offer to draw the currently ongoing game

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: User

Related FR: 2.5.4.

Pre-conditions:

+user is authorised

+the game exists

+user is one of the players in the game

Post-conditions:

+the other player is alerted that the opponent is offering a tie

+if the other player has offered a draw within the same move in the past 30 seconds, the game is tied

Request:

+gameId

Response:

+void

PUT /games/{id}/draw

Name: send a message to your opponent

Description: send a message out of a selection of pre-typed ones to your opponent

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: User

Related FR: 2.5.5.

Pre-conditions:

+user is authorised

+the game exists

+the message is valid

+user is one of the players of a game

Post-conditions:

+the other player receives the message

Request:

+gameId

+msgId

Response:

+void

POST /games/{id}/message

Name: view all games

Description: view all past and ongoing games of a certain format of a certain player

Operation Type: QUERY

Exposition: EXTERNAL

Roles: Admin

Related FR: 3.2.

Pre-conditions:

+admin is authorised

+the format is valid

+the player exists, if any

Post-conditions:

+the game list is returned

Request:

+maxShowAmount

+affiliationDistribution (black|white|random)

+timeTotal

+timeIncrement

+ranked

+playerId

Response:

+game[]

GET /games/

Name: view a user's info

Description: view a user's full info, including the normally-hidden details

Operation Type: QUERY

Exposition: EXTERNAL

Roles: Admin

Related FR: 3.3.

Pre-conditions:

+admin is authorised

+the user exists

Post-conditions:

+the user info is returned

Request:

+playerId

Response:

+info

GET /players/{id}/

Name: ban a User

Description: ban a User account

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: Admin

Related FR: 3.4.

Pre-conditions:

+admin is authorised

+the user exists

Post-conditions:

+the user is banned

Request:

+playerId

+reason

Response:

+void

POST /players/{id}/ban

Name: ban an IP

Description: ban a User IP

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: Admin

Related FR: 3.5.

Pre-conditions:

+admin is authorised

+the IP is not associated with any Admin accounts

Post-conditions:

+the IP is banned

Request:

+IP

+reason

Response:

+void

POST /ips/{id}/ban

Name: invalidate any game

Description: invalidate any game, compensating both players if they exist

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: Admin

Related FR: 3.6.

Pre-conditions:

+admin is authorised

+the game exists

+the game is not invalidated

Post-conditions:

+the game is set as invalidated

+the players' ratings are compensated

Request:

+gameId

+reason

Response:

+void

POST /games/{id}/invalidate

Name: delete any game

Description: delete any game

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: Admin

Related FR: 3.7.

Pre-conditions:

+admin is authorised

+the game exists

Post-conditions:

+the game is deleted

+the players are no longer in the game

Request:

+gameId

Response:

+void

DELETE /games/{id}/

Name: edit a User's rating

Description: edit a User's casual and ranked rating

Operation Type: COMMAND

Exposition: EXTERNAL

Roles: Admin

Related FR: 3.8.

Pre-conditions:

+admin is authorised

+the user exists

Post-conditions:

+the user's ratings are edited

Request:

+playerId

+casualRating

+rankedRating

Response:

+void

PATCH /players/{id}/edit