

ENDPOINT	URL	METHOD	INPUT	OUTPUT	COMMENTS
Users					
register	/users/	POST	{ user: string, pass: string }		
login	/users/login/	POST	{ user: string, pass: string }	{ token: string }	
Boards					
boards	/boards	GET		[{ id: integer, name: string }]	
Lobbies (waiting games)					
list lobbies	/rooms/	GET		[ROOM]	
create lobby	/rooms/	POST	{ name: string, board_id: integer }	ROOM	
get lobby	/rooms/<id>/	GET		ROOM	
start game	/rooms/<id>/	PATCH			PRE: 3 <= count_players <= 4 AND not started
cancel lobby	/rooms/<id>/	DELETE			
join lobby	/rooms/<id>/	PUT			
Games					
list all my games	/games/	GET		[{ id: integer, name: string, in_turn: USERNAME }]	
Public information					
game status	/games/<id>	GET		{ players: [PLAYER], robber: HEX_POSITION, current_turn: { user: USERNAME, dice: (int, int) }, winner: optional USERNAME, }	
board status	/games/<id>/board	GET		{ hexes: [HEX], }	
Personal information					
player personal info	/games/<id>/player	GET		{ resources: [RESOURCE], cards: [CARD_TYPE], }	

ENDPOINT	URL	METHOD	INPUT	OUTPUT	COMMENTS
play action	/games/<id>/player/actions	POST	{ type: ACTION, payload: PAYLOAD }		
player available actions	/games/<id>/player/actions	GET		[{ type: ACTION, payload: PAYLOAD }]	Un elemento por cada tipo de acción posible. Para cada tipo de acción, el payload indica las posiciones habilitadas (en caso de corresponderse).
Transactions					
get transaction status	/games/<id>/transactions	GET		{ offer: [RES], requested: [RES], responses: [{ player: USERNAME, answer: accepted declined }] }	
start transaction	/games/<id>/transactions	POST	{ offer: [RES], requested: [RES] }		PRE: usuario en turno
cancel transactions	/games/<id>/transactions	DELETE			PRE: usuario en turno
accept/decline	/games/<id>/transactions	PUT	{ action: accept decline, }		PRE: usuario no en turno
complete transaction	/games/<id>/transactions	PATCH	{ player: USERNAME, }		PRE: usuario en turno