



GET	POST	DELETE	GET
players	players	players	players
/ {id}	/	/ {id}	/ {id}
	{ name, color, aiFlag }		
[... player object ...]	{ name, color, ai, id }		{ name, color, ai, id }

GET	POST	DELETE	GET
games	games	games	games
/	/	/ {id}	/ {id}
			{ timestamp }
[... game object ...]	{ ... game object ... }		{ ... game object ... + turns }

GET	POST	DELETE
games	games	games
/ {id} / players	/ {id} / players	/ {id} / players
[... player object ...]		

Approaches to handling game progress:

- sockets
- polling: request game state, passing in last request timestamp and receiving back current state + game events; game then plays out events with delay until human players turn or out of events and then calls to get more events if the game is not finished

POST	POST
games	games
/ {id} / start	/ {id} / cols / {num}
	{ playerId }

Polling Approach:

- start game / take turn
- request game state, optionally passing in epoch of last known state (if no epoch is passed in, assume epoch is start)
- response contains full game state at given timestamp and a list of turns that have been taken as well as any pending turns (who its waiting on to take a turn)
- if turns have been taken, play those out with delay
- after playing out turns:
 - if it is user's turn, prompt them to take turn
 - if it is another players turn, indicate it's another players turn
 - if the game is over, display game over