Trello web application REST design

Constraint	Design in Trello
Client-server architecture	The client is Trello web application and the server is the backend that responds to the client responses
Layered system	No proof is available. It can be assumed that Trello follows client-server layered architecture
Stateless interactions	API requests include the information to grab all resources so that the server does not have to save information about user states Example: https://api.trello.com/1/boards/BdarzfKF/?fields=id&actions=addAttachmentToCard&actions_limit=2&action_fields=idMemberCreator&action_memberCreator_fields=fullName
Caching	Web browser saves login as cookies so that the user can stay logged in to make requests

Uniform interface for communication	API requests use URL.
	
	Example:
	https://api.trello.com/1/boards
	
	API responses return JSON objects
	
	Example:
	{
	"id": "54a17d76d4a5072e3931736b",
	"actions": [
	{
	"id": "54a1b7c3a6ea7c2b1eaa5cdf",
	"idMemberCreator": "53baf533e697a982248cd73f",
	"memberCreator": {
	"id": "53baf533e697a982248cd73f",
	"fullName": "Lauren Moon"
	}
	},
	{
	"id": "54a1b73f030916211e718516",
	"idMemberCreator": "53baf533e697a982248cd73f",
	"memberCreator": {
	"id": "53baf533e697a982248cd73f",
	"fullName": "Lauren Moon"
	}
	}
	]
	}