

Web Advanced API Documentation

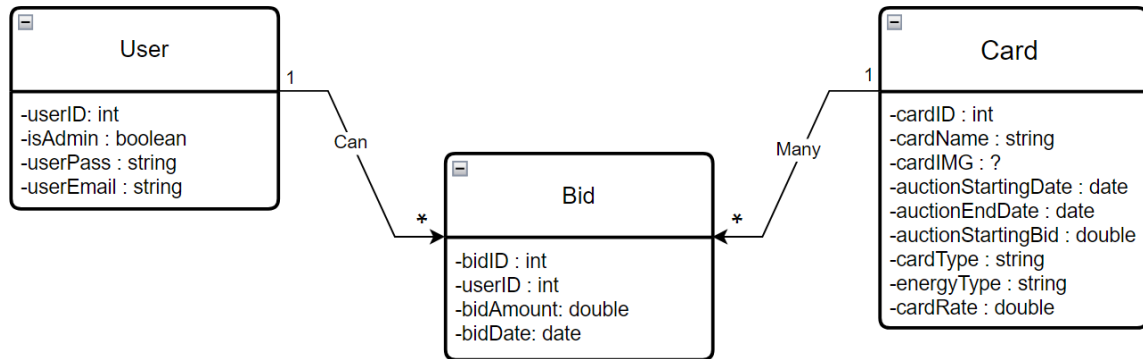
Table of Content

<i>Class diagram</i>	2
<i>GET requests</i>	3
<i>POST requests</i>	4
<i>PATCH requests</i>	6
<i>DELETE requests</i>	6

Class diagram

This is my class diagram with all my 3 entities (Users, Bids and Cards)

A User can place multiple Bids (1..*), but each Bid is associated with only one User (1). This relationship ensures that bids are assign to a specific user.
An Card can have multiple Bids (1..*), but each Bid is associated with only one Card (1). This relationship ensures that bids are connected to a specific Card.



GET requests

GET	users/me/cards needs token from user		
Gets all the won cards from the user needs token from user			
Parameters:	Name	Type	Description
* required			
Responses:	Code	Description / example if successful	
	200	Return a list of all won cards from the user	

GET	cards needs token from user		
Retrieve all Cards registered on the website needs token from user			
Parameters:	Name	Type	Description
<i>* required</i>		path	Display all cards
	cardName	query	Filter cards by name
	cardType	query	Filter cards by type
	energyType	query	Filter cards by energy
	cardRate	query	Filter cards by rate
Responses:	Code	Description / example if successful	
	200	Return a list of all Cards	

GET	cards/popular		
Retrieve the popular cards of the site (random 5 cards)			
Parameters:	Name	Type	Description
* required			Retrieve the popular cards of the site
Responses:	Code	Description / example if successful	
	200	Returns the User with the ID that was provided.	
	404	User ID is incorrect	

GET	cards/{cardID}		
needs token from user			
Retrieve the Card by ID			
Parameters:	Name	Type	Description
* required	cardID	path	Id of the card to get the information
Responses:	Code	Description / example if successful	
	200	Returns the Card Information with the ID that was provided.	
	404	Card ID is incorrect	

GET	cards/{cardD}/bids needs token from user		
Retrieve all bids from the selected Card by Its ID			
Parameters:	Name	Type	Description
* required	cardID	path	Id of the card to get bids
Responses:	Code	Description / example if successful	
	200	Return a list of all the bids from the selected Card	
	404	Card ID not found	

POST requests

POST	tokens/		
User will login with his credentials and will create a Token			
Parameters:	Name	Type	Description
* required	User	body	The User information needed to be in a JSON format to login (email,password)
Responses:	Code	Description / example if successful	
	200	User Login successfully, token created and given to user	
	404	Invalid input data - Email or Password not found	

POST	users		
A new User will be created/registred			
Parameters:	Name	Type	Description
* <i>required</i>	User	body	The user data to be added. Must be JSON with email and password. Example: { "email": "Rt01@gmail.com", "password": "Rt14" }
Responses:	Code	Description / example if successful	
	200	A new User will be created	
	404	Invalid input data - Email or Password not found	
	500	Internal server error with the error message.	

POST	cads/{cardID}/bids needs token from user and can't be admin		
A new Bid will be to the Card			
Parameters:	Name	Type	Description
* required	Card ID	path	Needs the Card ID to be added
	Bid	body	The user bid to be added. Must be

			JSON with bidAmount . Example: { "bidAmount": 22277 }
Responses:	Code	Description / example if successful	
	201	A new Bid will be created	
	400	Invalid input data (bid is less than the last higher bid, the user is an admin, the last bid is from the same user, the new bid can't be more than 30% of the last bid, auctions has not started, auction has ended)	
	404	Card not found	

POST	cards needs token from user and needs to be admin		
A new Card will be added			
Parameters:	Name	Type	Description
* required	Card	body	The card data to be added. Must be JSON with all card information. Example: { "cardName": "Alakazam", "cardImg": "aaa.png", "actionStartingDate": "14/12/2024", "auctionEndDate": "14/01/2025" "auctionStartingBid": 150.00, "cardType": "Pokemon", "energyType": "Psychic", "cardRate": 7.5, }
Responses:	Code	Description / example if successful	
	201	A new Card will be added	
	404	Invalid input data(not all fields are filled, auctionStartingBid < 20€ and auctionEndDate Is before the actionStartingDate)	

PATCH requests

PUT	cards/{cardID}		
needs token from user and to be admin			
Update the selected Card by its ID			
Parameters:	Name	Type	Description
* required	CardID	path	Needs the Card ID to be edit
	Card	body	The card data to be edited. Must be JSON with all card information. Example: { "cardName": "Alakazam", "cardImg": "aaa.png", "actionStartingDate": "14/12/2024", "auctionEndDate": "14/01/2025", "auctionStartingBid": 150.00, "cardType": "Pokemon", "energyType": "Psychic", "cardRate": 7.5, }
Responses:	Code	Description / example if successful	
	200	Card updated successfully	
	400	All errors 400 will be town here. (not all fields are filled, auctionStartingBid < 20€ and auctionEndDate Is before the actionStartingDate)	
	404	Card not found	

DELETE requests

DELETE	cards/{cardID}		
	needs token from user and to be admin		
Deletes the Card by his ID			
Parameters:	Name	Type	Description
* required	cardID	path	Needs the Card ID to be deleted
Responses:	Code	Description / example if successful	
	204	Card deleted successfully	
	404	Card not found	