Errors

Hide

Structural error at info
should have required property 'version'
missingProperty: version
<u>Jump to line 2</u>

Semantic error at paths./users.post.requestBody.content.application/json.schema.$ref
$ref paths must begin with `#/`
<u>Jump to line 30</u>

Semantic error at paths./users/{username}.put.requestBody.content.application/json.schema.$ref
$ref paths must begin with `#/`
<u>Jump to line 81</u>

Semantic error at paths./sessions.post.requestBody.content.application/json.schema.$ref
$ref paths must begin with `#/`
<u>Jump to line 102</u>

Monster Trading Cards Game

This is the specification of the required API endpoints for the MTCG server.

Servers

http://localhost:10001

Authorize 🔓

users   User registration and login   ⌃

POST   **/users**   Register a new user   ⌃

Register a new user with username and password

Parameters   Try it out

No parameters

Request body required   application/json

Example Value   Schema

```
{
  "Username": "kienboec",
  "Password": "daniel"
}
```

Responses

Code	Description	Links
201	User successfully created	*No links*
409	User with same username already registered	*No links*

GET   **/users/{username}**   Retrieves the user data for the given username.

Retrieves the user data for the username provided in the route. Only the admin or the matching user can successfully retrieve the data.

Parameters

Try it out

Name	Description
username * required string *(path)*	The username from a user's credentials. kienboec

Responses

Code	Description	Links
200	Data successfully retrieved	No links

Media type

```
application/json
```

Controls `Accept` header.

Example Value   Schema

```
{
  "Name": "Hoax",
  "Bio": "me playin...",
  "Image": ":-)"
}
```

Code	Description	Links
401	Access token is missing or invalid	No links
404	User not found.	No links

PUT   `/users/{username}`   Updates the user data for the given username.   ∧ 🔓

Retrieves the user data for the username provided in the route. Only the admin or the matching user can successfully retrieve the data.

Parameters   Try it out

Name	Description
username * required string *(path)*	The username from a user's credentials. kienboec

Request body required

application/json

Example Value   Schema

```
{
  "Name": "Hoax",
  "Bio": "me playin...",
  "Image": ":-)"
}
```

Responses

Code	Description	Links
200	User sucessfully updated.	*No links*
401	Access token is missing or invalid	*No links*

Code	Description	Links
404	User not found.	*No links*

POST **/sessions** Login with existing user ⌃

Login a user with username and password

Parameters [Try it out]

No parameters

Request body *required* [application/json]

Example Value Schema

```
{
  "Username": "kienboec",
  "Password": "daniel"
}
```

Responses

Code	Description	Links
200	User login successful	*No links*
401	Invalid username/password provided	*No links*

Media type

application/json

Example Value   Schema

```
"kienboec-mtcgToken"
```

packages   Package creation and acquisition   ⌃

POST **/packages**   Create new card packages (requires admin)   ⌃ 🔓

Creates a new package from an array of cards. Only the "admin" user may do this

Parameters   Try it out

No parameters

Request body ^{required}   application/json

Example Value   Schema

```json
[
  {
    "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "Name": "WaterGoblin",
    "Damage": 55
  },
  {
    "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "Name": "WaterGoblin",
    "Damage": 55
  },
  {
    "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "Name": "WaterGoblin",
    "Damage": 55
  },
  {
    "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "Name": "WaterGoblin",
    "Damage": 55
  },
  {
    "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
```

Responses

Code	Description	Links
201	Package and cards successfully created	*No links*
401	Access token is missing or invalid	*No links*

Code	Description	Links
403	Provided user is not "admin"	*No links*
409	At least one card in the packages already exists	*No links*

POST **/transactions/packages** Acquire a card package ∧ 🔓

Buys a card package with the money of the provided user

Parameters [Try it out]

No parameters

Responses

Code	Description	Links
200	A package has been successfully bought	*No links*

Media type

application/json

Controls *Accept* header.

Example Value Schema

Code	Description	Links

	``
	[
	  {
	    "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
	    "Name": "WaterGoblin",
	    "Damage": 55
	  }
	]
	``	

401	Access token is missing or invalid	*No links*
403	Not enough money for buying a card package	*No links*
404	No card package available for buying	*No links*

cards Card and deck management ∧

GET	**/cards** Shows a user's cards	∧ 🔓

Returns all cards that have been required by the provided user

Parameters   [Try it out]

No parameters

Responses

Code	Description	Links
200	The user has cards, the response contains these	*No links*

Media type

```
application/json
```

Controls `Accept` header.

Example Value   Schema

```json
[
  {
    "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "Name": "WaterGoblin",
    "Damage": 55
  }
]
```

Code	Description	Links
204	The request was fine, but the user doesn't have any cards	*No links*
401	Access token is missing or invalid	*No links*

GET   **/deck**   Shows the user's currently configured deck

Returns the cards that are owned by the uses and are put into the deck

Parameters   `Try it out`

Name	Description

format

string

(query)

The requested format of the response

Available values : plain, json

Default value : json

```
json
```

Responses

Code	Description	Links

200

The deck has cards, the response contains these

No links

Media type

```
application/json
```

Controls `Accept` header.

Example Value   Schema

```json
[
  {
    "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "Name": "WaterGoblin",
    "Damage": 55
  }
]
```

204

The request was fine, but the deck doesn't have any cards

No links

Code	Description	Links
401	Access token is missing or invalid	*No links*

PUT   **/deck**   Configures the deck with four provided cards   ∧ 🔓

Send four card IDs to configure a new full deck. A failed request will not change the previously defined deck.

Parameters   Try it out

No parameters

Request body ^{required}   application/json

Example Value   Schema

```
[
  "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "3fa85f64-5717-4562-b3fc-2c963f66afa6"
]
```

Responses

Code	Description	Links
200	The deck has been successfully configured	*No links*
400	The provided deck did not include the required amount of cards	*No links*
401	Access token is missing or invalid	*No links*
403	At least one of the provided cards does not belong to the user or is not available.	*No links*

game Battle related endpoints ^

GET **/stats** Retrieves the stats for an individual user ^ 🔓

Retrieves the stats for the requesting user.

Parameters   Try it out

No parameters

Responses

Code	Description	Links
200	The stats could be retrieved successfully.	No links

Media type

```
application/json
```

Controls `Accept` header.

Example Value   Schema

```
{
  "Name": "Hoax",
  "Elo": 0,
  "Wins": 0,
  "Losses": 0
}
```

| 401 | Access token is missing or invalid | No links |

GET   `/scoreboard`   Retrieves the user scoreboard ordered by the user's ELO.   ∧ 🔓

Parameters

[Try it out]

No parameters

Responses

Code	Description	Links
200	The scoreboard could be retrieved successfully.	No links

Media type

```
application/json
```

Controls `Accept` header.

Example Value   Schema

```
[
  {
    "Name": "Hoax",
    "Elo": 0,
    "Wins": 0,
    "Losses": 0
  }
]
```

Code	Description	Links
401	Access token is missing or invalid	No links

POST   **/battles**   Enters the lobby to start a battle.   ∧ 🔓

If there is already another user in the lobby, the battle will start immediately. Otherwise the request will wait for another user to enter.

Parameters   Try it out
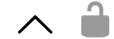
No parameters

Responses

Code	Description	Links
200	The battle has been carried out successfully.	*No links*

Media type

> **text/plain**

Controls Accept header.

Example Value   Schema

    string

Code	Description	Links
401	Access token is missing or invalid	*No links*

trading   Card trading   ∧

GET   **/tradings**   Retrieves the currently available trading deals.   ∧ 🔓

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	There are trading deals available, the response contains these	*No links*

Media type

```
application/json
```

Controls `Accept` header.

Example Value   Schema

```json
[
  {
    "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "CardToTrade": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "Type": "monster",
    "MinimumDamage": 15
  }
]
```

Code	Description	Links
204	The request was fine, but there are no trading deals available	*No links*
401	Access token is missing or invalid	*No links*

POST   **/tradings**   Creates a new trading deal.   ∧ 🔓

Creates a new trading deal. You can only create a deal for a card you own.

Parameters

Try it out

No parameters

Request body required

application/json

Example Value   Schema

```
{
  "Id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "CardToTrade": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "Type": "monster",
  "MinimumDamage": 15
}
```

Responses

Code	Description	Links
201	Trading deal successfully created	No links
401	Access token is missing or invalid	No links
403	The deal contains a card that is not owned by the user or locked in the deck.	No links

Code	Description	Links
409	A deal with this deal ID already exists.	*No links*

DELETE `/tradings/{tradingdealid}` Deletes an existing trading deal. ∧ 🔓

Deletes an existing trading deal. It is only allowed if the user owns the associated card.

Parameters   [**Try it out**]

Name	Description
tradingdealid * required string($uuid) *(path)*	The ID of the trading deal to delete. [tradingdealid]

Responses

Code	Description	Links
200	Trading deal successfully deleted	*No links*
401	Access token is missing or invalid	*No links*

Code	Description	Links
403	The deal contains a card that is not owned by the user.	*No links*
404	The provided deal ID was not found.	*No links*

POST **/tradings/{tradingdealid}** Carry out a trade for the deal with the provided card.

Carry out a trade for the deal with the provided card. Trading with self is not allowed.

Parameters

Try it out

Name	Description
tradingdealid * required `string($uuid)` *(path)*	The ID of the trading deal to carry out. `tradingdealid`

Request body

application/json

Example Value   Schema

```
"3fa85f64-5717-4562-b3fc-2c963f66afa6"
```

Responses

Code	Description	Links
200	Trading deal successfully executed.	*No links*
401	Access token is missing or invalid	*No links*
403	The offered card is not owned by the user, or the requirements are not met (Type, MinimumDamage), or the offered card is locked in the deck, or the user tries to trade with self	*No links*
404	The provided deal ID was not found.	*No links*

Schemas ⌃

UserCredentials ↵

UserData ↩

UserStats ↩

Card ↩

TradingDeal ↩