

# Aukcje Online API (1.0)

Download OpenAPI specification: [Download](#)

Autorzy:

Bartosz Ludwin, Filip Malejki, Mateusz Pawliczek

## auth

### Register

Rejestracja nowego użytkownika.

- sprawdzamy, czy e-mail nie jest zajęty,
- haszujemy hasło,
- zapisujemy do bazy,
- zwracamy nowe dane (bez hasła).

REQUEST BODY SCHEMA: `application/json`

<code>username</code> <code>required</code>	string (Username)
--	-------------------

<code>email</code> <code>required</code>	string <email> (Email)
---	------------------------

<code>password</code> <code>required</code>	string (Password) <code>&gt;= 6 characters</code>
--	---

### Responses

> **200** Successful Response

> **422** Validation Error

POST /register

## Request samples

### Payload

#### Content type

application/json

Copy

```
{
  "username": "JanKowalski",
  "email": "jan@example.com",
  "password": "mojeHaslo123"
}
```

## Response samples

200

422

#### Content type

application/json

Copy

```
{
  "username": "JanKowalski",
  "email": "jan@example.com",
  "id": "string",
  "role": "string"
}
```

## Login

Logowanie użytkownika. Używamy fieldów:

- username (przechowujemy tam email),
- password. Generujemy access i refresh token, zapisujemy refresh\_token w bazie.

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

grant_type >	Grant Type (string) or Grant Type (null) (Grant Type)
username required	string (Username)
password required	string (Password)
scope	string (Scope) Default: <input type="text" value=""/>
client_id >	Client Id (string) or Client Id (null) (Client Id)
client_secret >	Client Secret (string) or Client Secret (null) (Client Secret)

Responses

- > 200 Successful Response
- > 422 Validation Error

POST /login

Response samples

200

422

Content type  
application/json

Copy

```
{
  "access_token": "string",
  "refresh_token": "string",
  "token_type": "bearer"
}
```

# Refresh Access Token

Odświeżenie tokena: klient podaje refresh\_token. Sprawdzamy, czy token jest ważny i czy znajduje się w liście refresh\_tokens u użytkownika.

REQUEST BODY SCHEMA: application/json

<code>refresh_token</code> <i>required</i>	string (Refresh Token)
---	------------------------

## Responses

> **200** Successful Response

> **422** Validation Error

POST /token/refresh

## Request samples

Payload

Content type  
application/json

```
{  
  "refresh_token": "string"  
}
```

Copy

## Response samples

200

422

Content type  
application/json

```
{  
  "access_token": "string",  
  "refresh_token": "string",  
  "token_type": "bearer"  
}
```

---

## Logout

Wylogowanie z sesji

AUTHORIZATIONS: > *OAuth2PasswordBearer*

REQUEST BODY SCHEMA: application/json

---

refresh_token required	string (Refresh Token)
---------------------------	------------------------

---

## Responses

> **200** Successful Response

> **422** Validation Error

POST /logout

## Request samples

Payload

**Content type**  
application/json

```
{  
  "refresh_token": "string"  
}
```

## Response samples

**200****422**

Content type  
application/json

null

Copy

---

## users

### List Users

Lista wszystkich użytkowników - dostępne tylko dla administratora.

AUTHORIZATIONS: > *OAuth2PasswordBearer*

### Responses

> **200** Successful Response

GET /users/

## Response samples

200

Content type

application/json

Copy   Expand all   Collapse all

```
[
  - {
    "username": "JanKowalski",
    "email": "jan@example.com",
    "id": "string",
    "role": "string"
  }
]
```

# Get User Profile

Pobranie profilu konkretnego użytkownika. Admin może pobrać każdego użytkownika, zwykły user tylko siebie.

AUTHORIZATIONS: >   *OAuth2PasswordBearer*

PATH PARAMETERS

user_id	string (User Id)
required	

## Responses

> **200** Successful Response

> **422** Validation Error

GET   /users/{user\_id}

## Response samples

200

422

## Content type

application/json

Copy

```
{
  "username": "JanKowalski",
  "email": "jan@example.com",
  "id": "string",
  "role": "string"
}
```

## Get User Bids

Historia licytacji (ofert) dla danego użytkownika. Tylko właściciel konta lub administrator.

AUTHORIZATIONS: > *OAuth2PasswordBearer*

### PATH PARAMETERS

<code>user_id</code> <code>required</code>	string (User Id)
---	------------------

## Responses

> **200** Successful Response

> **422** Validation Error

GET /users/{user\_id}/bids

### Response samples

200

422



**Content type**

application/json

Copy

Expand all

Collapse all

```
[
  - {
    "id": "string",
    "auction_id": "string",
    "user_id": "string",
    "amount": 0,
    "timestamp": "2019-08-24T14:15:22Z"
  }
]
```

## auctions

### List Active Auctions

Lista wszystkich aktywnych aukcji.

### Responses

> **200** Successful Response

GET /auctions

### Response samples

**200**

## Content type

application/json

Copy

Expand all

Collapse all

```
[
  - {
    "id": "string",
    "title": "string",
    "description": "string",
    "owner_id": "string",
    "current_price": 0,
    "created_at": "2019-08-24T14:15:22Z"
  }
]
```

## Create Auction

Tworzenie nowej aukcji - tylko zalogowani użytkownicy.

- Ustawiamy owner\_id na \_id użytkownika w kolekcji.
- current\_price przyjmujemy jako starting\_price.

AUTHORIZATIONS: >*OAuth2PasswordBearer*

REQUEST BODY SCHEMA: application/json

title required	string (Title)
-------------------	----------------

description >	Description (string) or Description (null) (Description)
---------------	--

starting_price required	number (Starting Price) <input type="text" value="0"/>
----------------------------	--

## Responses

> **200** Successful Response> **422** Validation Error

POST /auctions

## Request samples

### Payload

**Content type**

application/json

Copy

```
{
  "title": "Laptop Lenovo",
  "description": "Używany, stan bardzo dobry",
  "starting_price": 100
}
```

## Response samples

**200****422****Content type**

application/json

Copy

```
{
  "id": "string",
  "title": "string",
  "description": "string",
  "owner_id": "string",
  "current_price": 0,
  "created_at": "2019-08-24T14:15:22Z"
}
```

## Get Auction

Szuka aukcji po podanym ID. Jeśli nie ma → 404.

PATH PARAMETERS

auction\_id  
required

string (Auction Id)

---

## Responses

> **200** Successful Response

> **422** Validation Error

GET /auctions/{auction\_id}

## Response samples

**200**

**422**

Content type

application/json

Copy

```
{
  "id": "string",
  "title": "string",
  "description": "string",
  "owner_id": "string",
  "current_price": 0,
  "created_at": "2019-08-24T14:15:22Z"
}
```

---

## Admin Edit Auction

Administrator edytuje dane aukcji. Dozwolone pola: title, description, starting\_price. Jeśli istnieją oferty, nie można zmienić starting\_price.

AUTHORIZATIONS: > *OAuth2PasswordBearer*

PATH PARAMETERS

---

auction\_id

required

string (Auction Id)

REQUEST BODY SCHEMA:

application/json

property name\*

additional property

any

Responses

> 200 Successful Response

> 422 Validation Error

PATCH /auctions/{auction\_id}

Request samples

Payload

Content type

application/json

{ }

Copy

Response samples

200

422

Content type

application/json

```
{
  "id": "string",
  "title": "string",
  "description": "string",
  "owner_id": "string",
  "current_price": 0,
```

Copy

```
"created_at": "2019-08-24T14:15:22Z"
}
```

## Place Bid

Składanie oferty na aukcję (podbicie bieżącej ceny). Warunki:

- Podana kwota > current\_price.
- Aukcja musi istnieć.
- Tylko zalogowani użytkownicy.

AUTHORIZATIONS: > *OAuth2PasswordBearer*

### PATH PARAMETERS

auction_id required	string (Auction Id)
------------------------	---------------------

REQUEST BODY SCHEMA: application/json

amount required	number (Amount) > 0
--------------------	---------------------

## Responses

> **200** Successful Response

> **422** Validation Error

POST /auctions/{auction\_id}/bid

## Request samples

Payload

Content type  
application/json

Copy

```
{
  "amount": 150
}
```

## Response samples

200

422

Content type  
application/json

Copy

```
{
  "id": "string",
  "auction_id": "string",
  "user_id": "string",
  "amount": 0,
  "timestamp": "2019-08-24T14:15:22Z"
}
```

## Close Auction

Zamykanie aukcji (przenosimy ją do historii). Tylko:

- właściciel aukcji (owner\_id) lub
- administrator.

AUTHORIZATIONS: > *OAuth2PasswordBearer*

PATH PARAMETERS

auction_id required	string (Auction Id)
------------------------	---------------------

## Responses

> 200 Successful Response

> **422 Validation Error**

POST    /auctions/{auction\_id}/close

## Response samples

200

422

Content type

application/json

null

Copy

## reports

## Auctions History

Pobranie historii wszystkich aukcji (dla administratora).

AUTHORIZATIONS: >    *OAuth2PasswordBearer*

## Responses

> **200 Successful Response**

GET    /reports/history



## Response samples

**200**

### Content type

application/json

Copy

Expand all

Collapse all

```
[
  - {
    "id": "string",
    "title": "string",
    "description": "string",
    "owner_id": "string",
    "created_at": "2019-08-24T14:15:22Z",
    "closed_at": "2019-08-24T14:15:22Z",
    "winner_id": "string",
    "final_price": 0
  }
]
```

## Get User Spending

Pobranie sumy wydatków każdego użytkownika, który wygrał przynajmniej jedną aukcję. Widoczne tylko dla administratora.

AUTHORIZATIONS: >*OAuth2PasswordBearer*

## Responses

> **200** Successful Response

GET /reports/user-spending

## Response samples

200

Content type

application/json

Copy

null

# Top Winners

Pobranie listy użytkowników z największą liczbą wygranych aukcji (domyślnie top 10). Widoczne tylko dla administratora.

AUTHORIZATIONS: > OAuth2PasswordBearer

## QUERY PARAMETERS

limit

integer (Limit)

Default: 10

## Responses

> 200 Successful Response

> 422 Validation Error

GET /reports/top-winners

## Response samples

200

422

Content type

application/json

Copy

null

---

## Total Cashflow

Pobranie całkowitej wartości pieniężnej wygenerowanej przez zakończone aukcje. Dostępne tylko dla administratora.

AUTHORIZATIONS: > *OAuth2PasswordBearer*

## Responses

> **200** Successful Response

GET /reports/total-cashflow

## Response samples

**200**

**Content type**

application/json

Copy

null

---

## High Value Auctions

Pobranie aukcji, których cena końcowa przekroczyła określoną wartość minimalną (domyślnie 1000). Widoczne tylko dla administratora.

AUTHORIZATIONS: > *OAuth2PasswordBearer*

QUERY PARAMETERS

min\_price

number (Min Price)  
Default:

Responses

- > 200 Successful Response
- > 422 Validation Error

GET /reports/high-value-auctions

Response samples

200

422

Content type  
application/json

Copy

null

Last Week Auctions

Pobranie aukcji utworzonych w ciągu ostatnich 7 dni. Widoczne tylko dla administratora.

AUTHORIZATIONS: > OAuth2PasswordBearer

Responses

- > 200 Successful Response

GET /reports/last-week-auctions

## Response samples

200

Content type

application/json

null

Copy

---

## Last Month Auctions

Pobranie aukcji utworzonych w ciągu ostatnich 30 dni. Widoczne tylko dla administratora.

AUTHORIZATIONS: > *OAuth2PasswordBearer*

## Responses

> 200 Successful Response

GET /reports/last-month-auctions

## Response samples

200

Content type

application/json

Copy

null

---

## Last 6H Auctions

Pobranie aukcji utworzonych w ciągu ostatnich 6 godzin. Widoczne tylko dla administratora.

AUTHORIZATIONS: > *OAuth2PasswordBearer*

## Responses

> **200** Successful Response

GET /reports/last-6h-auctions

## Response samples

**200**

Content type

application/json

Copy

null

---

## Auctions Stats

Pobranie statystyk aukcji: liczba aktywnych i zamkniętych aukcji. Widoczne tylko dla administratora.

AUTHORIZATIONS: > *OAuth2PasswordBearer*

## Responses

> **200** Successful Response

GET /reports/auctions-stats

### Response samples

**200**

Content type

application/json

Copy

null

---

## Root

Prosty root endpoint do sprawdzenia czy API żyje

## Responses

> **200** Successful Response

GET /

### Response samples

**200**

Content type

application/json

null