

Dokumentacja RESTful API - projekt "Doorime"

1. Tworzenie użytkownika:

- a. Ścieżka: **/create_user**
- b. Metoda: **POST**
- c. Wymagane dodatkowe pola:
 - i. Content-Type: application/json
 - ii. Content-Length: [length]
 - iii. JSON: `{ "username" : "[nazwa nowego użytkownika]", "password" : "[hasło]" }`
- d. cURL:

```
curl --location --request POST 'localhost:50000/create_user' \
--header 'Content-Type: application/json' \
--data-raw '{"username": "Test01", "password": "12345"}'
```
- e. HTTP:

```
POST /create_user HTTP/1.1
Host: localhost:50000
Content-Type: application/json

{"username": "Test01", "password": "12345"}
```
- f. Zwraca:
 - i. `{ 'message' : 'User already exist!' }` - username jest już zajęte (każdy username musi być unikalny)
 - ii. `{ 'message' : 'New user created!' }` - poprawne utworzenie użytkownika

2. Uwierzytelnianie:

- a. Ścieżka: **/login**
- b. Metoda: **GET**
- c. Wymagane dodatkowe pola:
 - i. Pole 'Basic Auth' zawierające: `base64('username:password')`
- d. cURL:

```
curl --location --request GET 'localhost:50000/login' \
--header 'Authorization: Basic VGVzdDxOjEyMzQ1' \
--data-raw ''
```
- e. HTTP:

```
GET /login HTTP/1.1
Host: localhost:50000
Authorization: Basic VGVzdDxOjEyMzQ1
```
- f. Zwraca:
 - i. 200 - JSON `{ "message" : "ok", "token" : "[token]" }` - w przypadku poprawnego zalogowania
 - ii. `{ 'message' : 'Login and password required!' }` - jeśli nie podana jednego z wymaganych parametrów
 - iii. 200 - `{ 'message' : 'User doesn't exist!' }` - brak użytkownika w bazie
 - iv. 200 - `{ 'message' : 'Login and password don't match!' }` - niepoprawne logowanie

3. Tworzenie kolekcji i przypisywanie jej użytkownikowi (nowe kolekcje można przypisywać tylko sobie):

a. Ścieżka: **/add_collection**

b. Metoda: **PUT**

c. Wymagane dodatkowe pola:

i. Content-Type: application/json

ii. Content-Length: [length]

iii. x-access-token: [token]

iv. JSON: {"collection": "[id kolekcji]", "name": "[nazwa kolekcji]"}

d. cURL

```
curl --location --request PUT 'localhost:50000/add_collection' \
--header 'x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IiRlc3QwMSIsImV
4cCI6MTU4NjYzMzEyMH0.gtlClqdiCGTfGWZQq2xwiTbhqtYtMtJL-d_EGFKC4G
E' \
--header 'Content-Type: application/json' \
--data-raw '{"collection": "Grove Street"}
```

e. HTTP:

PUT /add_collection HTTP/1.1

Host: localhost:50000

x-access-token:

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IiRlc3QwMSIsImV
4cCI6MTU4NjYzMzEyMH0.gtlClqdiCGTfGWZQq2xwiTbhqtYtMtJL-d_EGFKC4G
E

Content-Type: application/json

{"collection": "Grove Street"}

f. Zwraca:

i. 200 - {'message': 'collection already in use!'} - kolekcja o podanym id już istnieje

ii. 200 - {'message': 'collection assigned correctly!'} - operacja zakończona poprawnie

iii. 401 - {'message': 'Token is missing!'} - brak tokenu

iv. 401 - {'message': 'Token is invalid!'} - niepoprawny token (wystąpi również dla przedawnionego tokena)

g. Uwagi:

i. ID kolekcji jest stringiem i musi być unikalne dla całej bazy

ii. Nazwa kolekcji również jest stringiem, jednak jest opcjonalna. Jeśli nie zostanie wprowadzona przyjmuje domyślną wartość "My collection"

4. Dodawanie urządzenia do kolekcji:

a. Ścieżka **/add_device**

b. Metoda: **PUT**

c. Wymagane dodatkowe pola:

- i. Content-Type: application/json
- ii. Content-Length: [length]
- iii. x-access-token: [token]
- iv. JSON: {"collection" : "[nazwa kolekcji już przypisanej do użytkownika]", "device" : "[nazwa urządzenia]"}"
- v. **ALTERNATYWNI** zamiast "iv": JSON: {"device" : "[id urządzenia]", "name" : "[nazwa urządzenia]"}

d. cURL:

```
curl --location --request PUT 'localhost:50000/add_device' \
--header 'x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IiRlc3QwMSIsImV
4cCI6MTU4NjYzMzEyMH0.gtlClqdiCGTfGWZQq2xwiTbhqtYtMtJL-d_EGFKC4G
E' \
--header 'Content-Type: application/json' \
--data-raw '{"collection" : "Grove Street", "device" : "Home"}'
```

e. HTTP:

```
PUT /add_device HTTP/1.1
Host: localhost:50000
x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IiRlc3QwMSIsImV
4cCI6MTU4NjYzMzEyMH0.gtlClqdiCGTfGWZQq2xwiTbhqtYtMtJL-d_EGFKC4G
E
Content-Type: application/json

{"collection" : "Grove Street", "device" : "Home"}
```

f. Zwraca:

- i. 200 - {'message' : 'you don't manage this collection!'} - Kolekcja istnieje, ale jest przypisana do innego użytkownika
- ii. 200 - {'message' : 'device already assigned to another collection!'} - urządzenie już zostało przypisane do konkretnej kolekcji (nazwy urządzeń są unikalne. Konwencja na nazwę urządzeń, to MAC address ESP, do którego są podpięte kontaktry)
- iii. 200 - {'message' : 'device added to collection!'} - operacja zakończona pomyślnie
- iv. 401 - {'message' : 'Token is missing!'} - brak tokenu
- v. 401 - {'message' : 'Token is invalid!'} - niepoprawny token (wystąpi również dla przedawnionego tokena)

5. Wyświetlanie informacji o wszystkich urządzeniach, przypisanych do wszystkich kolekcji danego użytkownika:

a. Ścieżka: **/get_all_data**

- b. Metoda: **GET**
- c. Wymagane dodatkowe pola:
 - i. x-access-token: [token]
- d. cURL:


```
curl --location --request GET 'localhost:50000/get_all_data' \
--header 'x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IiRlc3QwMSIsImV
4cCI6MTU4NjYzMzEyMH0.gtlClqdiCGTfGWZQq2xwiTbhqtYtMtJL-d_EGFKC4G
E' \
--data-raw ''
```
- e. HTTP:


```
GET /get_all_data HTTP/1.1
Host: localhost:50000
x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IiRlc3QwMSIsImV
4cCI6MTU4NjYzMzEyMH0.gtlClqdiCGTfGWZQq2xwiTbhqtYtMtJL-d_EGFKC4G
E
```
- f. Zwraca:
 - i. 200 - {"message": "ok", "result": {[JSON z danymi]}}
 - ii. 401 - {'message': 'Token is missing!'} - brak tokenu
 - iii. 401 - {'message': 'Token is invalid!'} - niepoprawny token (wystąpi również dla przedawnionego tokena)

6. Weryfikacja poprawności tokena:

- a. Ścieżka: **/validate**
- b. Metoda: **GET**
- c. Wymagane dodatkowe pola:
 - i. x-access-token: [token]
- d. cURL:


```
curl --location --request GET 'localhost:50000/validate' \
--header 'x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw
ljoXNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA' \
--data-raw ''
```
- e. HTTP:


```
GET /validate HTTP/1.1
Host: localhost:50000
x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw
ljoXNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA
```
- f. Zwraca:

- i. 200 - {"message" : "ok", "Token expiration date" : "[dd.mm.yyyy hh:mm:ss]"}
- dla poprawnego tokenu
- ii. 401 - {'message' : 'Token is missing!'} - brak tokenu
- iii. 401 - {'message' : 'Token is invalid!'} - niepoprawny token (wystąpi również dla przedawnionego tokena)

7. Zmiana nazwy kolekcji:

- a. Ścieżka: **/rename_collection**
- b. Metoda: **PUT**
- c. Wymagane dodatkowe pola:
 - i. Content-Type: application/json
 - ii. Content-Length: [length]
 - iii. x-access-token: [token]
 - iv. JSON: {"collection" : "[nazwa kolekcji już przypisanej do użytkownika]", "new name" : "[nowa nazwa]"}
- d. cURL:


```
curl --location --request PUT 'localhost:50000/rename_collection' \
--header 'x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw
ljoXNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA' \
--header 'Content-Type: application/json' \
--data-raw '{"collection" : "B9", "new name" : ""}'
```
- e. HTTP:


```
PUT /rename_collection HTTP/1.1
Host: localhost:50000
x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw
ljoXNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA
Content-Type: application/json

{"collection" : "B9", "new name" : ""}
```
- f. Zwraca:
 - i. 200 - {'message' : 'invalid argument set!'} - jeśli w zapytaniu nie znajdują się jednocześnie "collection" i "new name"
 - ii. 200 - {'message' : 'collection name updated successfully!'} - operacja zakończona powodzeniem
 - iii. 200 - {'message' : 'operation not permitted!'} - w innym przypadku (na przykład jeśli kolekcja o wskazanym id nie istnieje, lub nie jest zarządzana przez użytkownika)
 - iv. 401 - {'message' : 'Token is missing!'} - brak tokenu
 - v. 401 - {'message' : 'Token is invalid!'} - niepoprawny token (wystąpi również dla przedawnionego tokena)

8. Zmiana nazwy urządzenia:

- a. Ścieżka: **/rename_device**
- b. Metoda: **PUT**
- c. Wymagane dodatkowe pola:
 - i. Content-Type: application/json
 - ii. Content-Length: [length]
 - iii. x-access-token: [token]
 - iv. JSON: {"device": "[id urządzenia]", "new name": "[nowa nazwa]"}
- d. cURL:

```
curl --location --request PUT 'localhost:50000/rename_device' \
--header 'x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw
ljoXNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA' \
--header 'Content-Type: application/json' \
--data-raw '{"device": "AA:BB:CC:DD:EE:FF", "new name": "[nazwa
urządzenia]"}
```
- e. HTTP:

```
PUT /rename_device HTTP/1.1
Host: localhost:50000
x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw
ljoXNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA
Content-Type: application/json

{"device": "AA:BB:CC:DD:EE:FF", "new name": "[nazwa urządzenia]"}
```
- f. Zwraca:
 - i. 200 - {'message': 'invalid argument set!'} - jeśli w zapytaniu nie znajdują się jednocześnie "device" i "new name"
 - ii. 200 - {'message': 'device name updated successfully!'} - operacja zakończona powodzeniem
 - iii. 200 - {'message': 'operation not permitted!'} - w innym przypadku (na przykład jeśli urządzenie o wskazanym id nie istnieje, lub nie należy do kolekcji zarządzanej przez danego użytkownika)
 - iv. 401 - {'message': 'Token is missing!'} - brak tokenu
 - v. 401 - {'message': 'Token is invalid!'} - niepoprawny token (wystąpi również dla przedawnionego tokena)

9. Dodawanie ssid do listy sieci wifi:

- a. Ścieżka: **/add_wifi**
- b. Metoda: **PUT**
- c. Wymagane dodatkowe pola:
 - i. Content-Type: application/json
 - ii. Content-Length: [length]
 - iii. x-access-token: [token]
 - iv. JSON: {"ssid": "[nazwa sieci wlan]"}
- d. cURL:

```
curl --location --request PUT 'localhost:50000/add_wifi' \
```

```
--header 'x-access-token:  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw  
ljoxNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA' \  
--header 'Content-Type: application/json' \  
--data-raw '{"ssid" : "AGH Guest"}'
```

e. HTTP:

```
PUT /add_wifi HTTP/1.1  
Host: localhost:50000  
x-access-token:  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw  
ljoxNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA  
Content-Type: application/json
```

```
 '{"ssid" : "AGH Guest"}'
```

f. Zwraca:

- i. 200 - {'message' : 'invalid argument set!'} - jeśli w zapytaniu nie ma jsona, zawierającego "ssid"
- ii. 200 - {'message' : 'ssid added successfully!'} - operacja zakończona pomyślnie
- iii. 401 - {'message' : 'Token is missing!'} - brak tokenu
- iv. 401 - {'message' : 'Token is invalid!'} - niepoprawny token (wystąpi również dla przedawnionego tokena)

g. Uwagi:

- i. Dodanie ponownie już przypisanej sieci wifi do listy, zwróci informację o pozytywnym zakończeniu operacji, jednak nazwa nie zostanie ponownie dodana.

10. Usuwanie ssid z listy sieci wifi:

- a. Ścieżka: **/delete_wifi**
- b. Metoda: **DELETE**
- c. Wymagane dodatkowe pola:
 - i. Content-Type: application/json
 - ii. Content-Length: [length]
 - iii. x-access-token: [token]
 - iv. JSON: {"ssid" : "[nazwa sieci wlan]"}
- d. cURL:

```
curl --location --request DELETE 'localhost:50000/delete_wifi' \  
--header 'x-access-token:  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw  
ljoxNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA' \  
--header 'Content-Type: application/json' \  
--data-raw '{"ssid" : "AGH WPA"}'
```

e. HTTP:

```
DELETE /delete_wifi HTTP/1.1
```

Host: localhost:50000

x-access-token:

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhwIjoxNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA

Content-Type: application/json

{"ssid" : "AGH WPA"}

f. Zwraca:

- i. 200 - {'message' : 'invalid argument set!'} - jeśli w zapytaniu nie ma jsona, zawierającego "ssid"
- ii. 200 - {'message' : 'ssid deleted successfully!'} - operacja zakończona pomyślnie
- iii. 200 - {'message' : 'Failed, SSID does not exist!'} - Użytkownik nie miał na liście sieci wifi o wskazanym ssid
- iv. 401 - {'message' : 'Token is missing!'} - brak tokenu
- v. 401 - {'message' : 'Token is invalid!'} - niepoprawny token (wystąpi również dla przedawnionego tokena)

11. Usuwanie urządzenia z kolekcji:

a. Ścieżka: **/delete_device**

b. Metoda: **DELETE**

c. Wymagane dodatkowe pola:

- i. Content-Type: application/json
- ii. Content-Length: [length]
- iii. x-access-token: [token]
- iv. JSON: {"device" : "[id urządzenia]"}

d. cURL:

```
curl --location --request DELETE 'localhost:50000/delete_device' \
--header 'x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhwIjoxNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA' \
--header 'Content-Type: application/json' \
--data-raw '{"device" : "00:50:56:00:22:03"}'
```

e. HTTP:

DELETE /delete_device HTTP/1.1

Host: localhost:50000

x-access-token:

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhwIjoxNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA

Content-Type: application/json

{"device" : "00:50:56:00:22:03"}

f. Zwraca:

- i. 200 - {'message' : 'invalid argument set!'} - jeśli w zapytaniu nie znajdują się "device"
- ii. 200 - {'message' : 'device name deleted successfully!'} - operacja zakończona powodzeniem
- iii. 200 - {'message' : 'operation not permitted!'} - w innym przypadku (na przykład jeśli urządzenie o wskazanym id nie istnieje, lub nie należy do kolekcji zarządzanej przez danego użytkownika)
- iv. 401 - {'message' : 'Token is missing!'} - brak tokenu
- v. 401 - {'message' : 'Token is invalid!'} - niepoprawny token (wystąpi również dla przedawnionego tokena)
- g. Uwagi:
 - i. Możliwe jest usuwanie tylko jednego urządzenia na raz

12. Usuwanie pustej kolekcji:

- a. Ścieżka: **/delete_collection**
- b. Metoda: **DELETE**
- c. Wymagane dodatkowe pola:
 - i. Content-Type: application/json
 - ii. Content-Length: [length]
 - iii. x-access-token: [token]
 - iv. JSON: {"collection" : "[nazwa kolekcji już przypisanej do użytkownika]"}
- d. cURL:


```
curl --location --request DELETE 'localhost:50000/delete_collection' \
--header 'x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw
ljoXNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA' \
--header 'Content-Type: application/json' \
--data-raw '{"collection" : "D5"}'
```
- e. HTTP:


```
DELETE /delete_collection HTTP/1.1
Host: localhost:50000
x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw
ljoXNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA
Content-Type: application/json

{"collection" : "D5"}
```
- f. Zwraca:
 - i. 200 - {'message' : 'invalid argument set!'} - jeśli w zapytaniu nie znajduje się pole "collection"
 - ii. 200 - {'message' : 'operation not permitted!'} - jeśli kolekcja o podanym id nie istnieje, lub jeśli nie jest zarządzana przez danego użytkownika

- iii. `{'message' : 'collection dropped successfully!'}` - jeśli istnieje inny użytkownik, który ma przypisaną daną kolekcję. Wtedy użytkownik wywołujący tą metodę usunie kolekcję jedynie ze swojego konta.
- iv. `200 - {'message' : 'this collection is not empty!'}` - jeśli do kolekcji są przypisane jakiekolwiek urządzenia oraz użytkownik wywołujący metodę jest jedynym posiadaczem kolekcji. W takiej sytuacji konieczne jest przekazanie kolekcji innemu użytkownikowi za pomocą `/share_collection` i ponowne wywołanie metody. Innym rozwiązaniem jest ręczne usunięcie wszystkich urządzeń wchodzących w skład kolekcji i ponowne wywołanie `/delete_collection`
- v. `200 - {'message' : 'collection deleted successfully!'}` - Pusta kolekcja została bezpowrotnie usunięta. Pojawienie się komunikatu oznacza, że żaden inny użytkownik, niż wywołujący metodę, nie miał przypisanej usuwanej kolekcji
- vi. `401 - {'message' : 'Token is missing!'}` - brak tokenu
- vii. `401 - {'message' : 'Token is invalid!'}` - niepoprawny token (wystąpi również dla przedawnionego tokena)

13. Usuwanie użytkownika:

- a. Ścieżka: **`/delete_user`**
- b. Metoda: **`DELETE`**
- c. Wymagane dodatkowe pola:
 - i. `x-access-token: [token]`
- d. cURL:


```
curl --location --request DELETE 'localhost:50000/delete_user' \
--header 'x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IiBhd2VcdTAxNDli
LCJleHAiOiJlODY4MTY0MDB9.0hOkx7OH1hATlqSqxT1I-YhGhZrGqdpUFmCN
wWpYMA8'
```
- e. HTTP:


```
DELETE /delete_user HTTP/1.1
Host: localhost:50000
x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IiBhd2VcdTAxNDli
LCJleHAiOiJlODY4MTY0MDB9.0hOkx7OH1hATlqSqxT1I-YhGhZrGqdpUFmCN
wWpYMA8
```
- f. Zwraca:
 - i. `200 - {'message' : 'Failed, there is a collection assigned to the user!'}` - Jeśli istnieje jakakolwiek kolekcja, która jest przypisana do użytkownika
 - ii. `200 - {'message' : 'The user has been deleted!'}` - Użytkownik usunięty pomyślnie
 - iii. `401 - {'message' : 'Token is missing!'}` - brak tokenu
 - iv. `401 - {'message' : 'Token is invalid!'}` - niepoprawny token (wystąpi również dla przedawnionego tokena)

- g. Uwagi:
 - i. Możliwe jest usunięcie jedynie użytkownika, który nie posiada żadnej kolekcji. Jeśli chcemy usunąć użytkownika, który posiada kolekcję, trzeba najpierw wykonać /delete_collection na wszystkich posiadanych kolekcjach
 - ii. **Usunięcie użytkownika usuwa bezpowrotnie wpis z bazy danych sqlite, zawierający: ID użytkownika, username oraz hash hasła**

14. Udostępnianie kolekcji innemu użytkownikowi:

- a. Ścieżka: **/share_collection**
- b. Metoda: **POST**
- c. Wymagane dodatkowe pola:
 - i. Content-Type: application/json
 - ii. Content-Length: [length]
 - iii. x-access-token: [token]
 - iv. JSON: {"collection" : "[id kolekcji]", "successor" : "[username]"}
- d. cURL:


```
curl --location --request POST 'localhost:50000/share_collection' \
--header 'x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw
ljoXNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA' \
--header 'Content-Type: application/json' \
--data-raw '{"collection" : "2", "successor" : "Paweł"}'
```
- e. HTTP:


```
POST /share_collection HTTP/1.1
Host: localhost:50000
x-access-token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFkbWluliwiZXhw
ljoXNTkyNzI1NjM2fQ.jrCQpoEpOlviIZAYn6_ZGNoi1wGroZ2CDWMXxr1ySeA
Content-Type: application/json

{"collection" : "2", "successor" : "Paweł"}
```
- f. Zwraca:
 - i. {'message' : 'invalid argument set!'} - jeśli w body http brakuje jsona, zawierającego pole "collection" oraz "successor"
 - ii. {'message' : 'operation not permitted!'} - jeśli wskazana kolekcja nie istnieje, lub jeśli nie jest zarządzana przez użytkownika, który wywołał metodę
 - iii. {'message' : 'successor does not exist!'} - jeśli użytkownik, któremu chcemy udostępnić kolekcję nie istnieje
 - iv. {'message' : 'collection shared successfully'} - operacja zakończona pomyślnie