

# Podział na moduły

Kod aplikacji został podzielony na cztery moduły. Ze względu na stosowanie architektury klient-serwer jednym z modułów jest warstwa klienta (frontend), a pozostałe trzy moduły znajdują się w warstwie serwera (backend). Część serwerowa została podzielona na moduł czatu, moduł użytkowników oraz moduł ofert w celu łatwiejszego podziału pracy między programistów. Moduły serwerowe nie komunikują się między sobą, natomiast warstwa klienta komunikuje się z warstwą serwerową za pomocą REST API.

# Szczegółowa dokumentacja REST API

## Struktury danych

Wszystkie obiekty są reprezentowane w formacie JSON. Pola, w których typie zawarta jest fraza “nie wymagane przy wysyłaniu” nie powinny być wysyłane przez warstwę kliencką na serwer, natomiast zawsze będą zwrócone przez serwer. Pola te są automatycznie generowane przez serwer przy tworzeniu nowych obiektów.

## Ocena

Obiekt reprezentujący ocenę użytkownika.

Pola oceny:

- rating (int): ocena użytkownika w skali od 1 do 5
- rated\_user (int): identyfikator użytkownika, który jest oceniany

## Użytkownik

Obiekt reprezentujący użytkownika aplikacji.

Pola użytkownika:

- id (int, nie wymagane przy wysyłaniu): identyfikator użytkownika
- username (string): nazwa użytkownika
- email (string): email użytkownika
- user\_rating (float, nie wymagane przy wysyłaniu): średnia ocena użytkownika

## Kategoria

Obiekt reprezentujący gatunek książki.

Pola kategorii:

- id (int): identyfikator gatunku
- name (string): nazwa gatunku

## Stan

Obiekt reprezentujący stan książki.

Pola stanu:

- id (int): identyfikator stanu książki
- condition (string): nazwa stanu książki

## Lokalizacja

Obiekt reprezentujący lokalizację.

Pola lokalizacji:

- id (int): identyfikator lokalizacji
- name (string): nazwa lokalizacji

## Oferta

Obiekt reprezentujący ofertę.

Pola oferty:

- id (int, nie wymagane przy wysyłaniu): identyfikator oferty
- user (użytkownik, nie wymagane przy wysyłaniu): właściciel oferty
- category (kategoria): gatunek książki, której dotyczy oferta
- condition (stan): stan książki, której dotyczy oferta
- location (lokalizacja): lokalizacja, której dotyczy oferta
- images ([string], nie wymagane przy wysyłaniu): tablica adresów url zdjęć do oferty
- date (string, nie wymagane przy wysyłaniu): data zamieszczenia oferty
- title (string): tytuł książki, której dotyczy oferta
- author (string): autor książki, której dotyczy oferta
- description (string): opis oferty
- is\_deleted (boolean, nie wymagane przy wysyłaniu): aktualność oferty

## Wiadomość

Obiekt reprezentujący wiadomość.

Pola wiadomości:

- offer (int): identyfikator oferty, do której została wysłana wiadomość
- sender (użytkownik, nie wymagane przy wysyłaniu): użytkownik wysyłający wiadomość
- recipient (użytkownik): użytkownik będący adresatem wiadomości
- message (string): treść wiadomości
- sending\_time (string, nie wymagane przy wysyłaniu): czas wysłania wiadomości

## Komentarz

Obiekt reprezentujący komentarz w profilu użytkownika.

Pola komentarza:

- user (użytkownik): użytkownik, którego dotyczy komentarz
- author (użytkownik, nie wymagane przy wysyłaniu): autor komentarza
- content (string): treść komentarza
- adding\_time (string, nie wymagane przy wysyłaniu): czas dodania komentarza

## Odpowiedź

Obiekt reprezentujący odpowiedź od serwera na zapytanie.

Pola odpowiedzi:

- status (string): status wykonania zapytania
- desc (string, nie wymagane przy wysyłaniu): opis w przypadku błędu

## Moduł czatu

### (POST) Wysyłanie wiadomości

Punkt dostępowy pozwalający na wysyłanie wiadomości.

Adres url: /chat/message

Ciało zapytania:

- payload (wiadomość): zawiera wymagane pola obiektu wiadomość

Możliwe odpowiedzi:

- status code 201 - udało się wysłać wiadomość
- status code 409 - wiadomość nie spełnia wymogów bazy danych
- status code 400 - serwer nie mógł przetworzyć zapytania
- status code 422 - błędy w obiekcie wiadomości

### (GET) Lista ofert

Punkt dostępowy pozwalający na pobranie listy ofert w ramach których czatowano.

Adres url: /chat/getChats/

Możliwe odpowiedzi:

- status code 200 - zawiera tablice obiektów oferta
- status code 403 - brak uprawnień do pobrania listy ofert

### (GET) Lista czatów do oferty

Punkt dostępowy pozwalający na pobranie listy użytkowników z którymi czatowano w ramach oferty.

Adres url: /chat/getChats/<int:offer\_id>

- offer\_id - identyfikator oferty, której dotyczą czaty

Możliwe odpowiedzi:

- status code 200 - zawiera tablice obiektów użytkownik
- status code 403 - brak uprawnień do pobrania listy użytkowników

### (GET) Pobieranie wiadomości

Punkt dostępowy pozwalający na pobranie wiadomości między użytkownikami w ramach oferty.

Adres url: /chat/getChats/<int:offer\_id>/<int:user\_id>

- offer\_id - identyfikator oferty, której dotyczą wiadomości
- user\_id - identyfikator użytkownika, którego dotyczą wiadomości

Możliwe odpowiedzi:

- status code 200 - zawiera tablice obiektów wiadomości
- status code 403 - brak uprawnień do pobrania wiadomości

## Moduł ofert

### (GET) Pobieranie ofert

Punkt dostępowy pozwalający na pobranie ofert spełniających warunki filtrowania.

Adres url: /offers?category=&condition=&localization=&author=&title=&sort=

- category (int, opcjonalne) - identyfikator gatunku, po którym filtrujemy
- condition (int, opcjonalne) - identyfikator stanu książki, po którym filtrujemy
- localization (int, opcjonalne) - identyfikator lokalizacji, po którym filtrujemy
- author (string, opcjonalne) - wyszukiwanie po autorze książki
- title (string, opcjonalne) - wyszukiwanie po tytule książki
- sort (string, opcjonalne) - sortowanie wyniku. Powinien przyjmować wartości:
  - "title" - sortowanie po tytule
  - "-title" - sortowanie po tytule malejąco
  - "author" - sortowanie po autorze
  - "-author" - sortowanie po autorze malejąco
  - "date" - sortowanie po dacie dodania
  - "-date" - sortowanie po dacie dodania malejąco

Możliwe odpowiedzi:

- status code 200 - zawiera tablice obiektów oferta spełniających warunki filtrowania

### (GET) Pobieranie oferty

Punkt dostępowy pozwalający na pobranie oferty.

Adres url: /offers/<int:offer\_id>

- offer\_id - identyfikator oferty, którą chcemy przeglądać

Możliwe odpowiedzi:

- status code 200 - zawiera obiekt oferta
- status code 404 - oferta nie została odnaleziona

### (POST) Stworzenie oferty

Punkt dostępowy pozwalający na stworzenie oferty.

Adres url: /offers/create

Ciało zapytania:

- payload (oferta): zawiera wymagane pola obiektu oferta
- file (pliki, opcjonalne): zdjęcia dołączone do oferty, maksymalnie 5

Możliwe odpowiedzi:

- status code 201 - udało się utworzyć ofertę
- status code 400 - serwer nie mógł przetworzyć zapytania
- status code 403 - brak uprawnień do stworzenia oferty
- status code 422 - błędy w obiekcie oferty

## (PATCH) Edycja oferty

Punkt dostępowy pozwalający na edycję oferty.

Adres url: /offers/edit/<int:offer\_id>

Ciało zapytania:

- payload (oferta): zawiera wymagane pola obiektu oferta

Możliwe odpowiedzi:

- status code 200 - udało się zmienić dane
- status code 400 - serwer nie mógł przetworzyć zapytania
- status code 403 - brak uprawnień do edycji oferty
- status code 404 - nie znaleziono oferty
- status code 422 - błędy w obiekcie oferta

## (GET) Pobieranie ofert użytkownika

Punkt dostępowy pozwalający na pobranie ofert danego użytkownika.

Adres url: /offers/user/<int:user\_id>

- user\_id - identyfikator użytkownika, którego oferty chcemy pobrać

Możliwe odpowiedzi:

- status code 200 - zawiera tablicę obiektów oferta

## (DELETE) Usunięcie oferty

Punkt dostępowy pozwalający na usuwanie własnych ofert.

Adres url: /offers/delete/<int:offer\_id>

- offer\_id - identyfikator oferty, którą chcemy usunąć

Możliwe odpowiedzi:

- status code 200 - udało się usunąć ofertę
- status code 403 - brak uprawnień do usunięcia oferty
- status code 404 - nie znaleziono oferty

## (GET) Pobieranie stanów książek

Punkt dostępowy pozwalający na pobranie wszystkich możliwych stanów książek.

Adres url: /offers/conditions

Możliwe odpowiedzi:

- status code 200 - zawiera tablicę obiektów stan

## (GET) Pobieranie lokalizacji

Punkt dostępowy pozwalający na pobranie wszystkich możliwych lokalizacji.

Adres url: /offers/localizations

Możliwe odpowiedzi:

- status code 200 - zawiera tablicę obiektów lokalizacja

## (GET) Pobieranie gatunków

Punkt dostępowy pozwalający na pobranie wszystkich możliwych gatunków książek.

Adres url: /offers/categories

Możliwe odpowiedzi:

- status code 200 - zawiera tablicę obiektów kategoria

## Moduł użytkowników

### (POST) Zalogowanie się

Punkt dostępowy pozwalający na zalogowanie się użytkownika.

Adres url: /api/login/

Ciało zapytania:

- payload (json object): obiekt zawierający pola logowania (email i password)

Możliwe odpowiedzi:

- status code 200 - udało się zalogować
- status code 401 - błędne dane logowania

### (POST) Rejestracja

Punkt dostępowy pozwalający na zarejestrowanie się użytkownika.

Adres url: /api/register/

Ciało zapytania:

- payload (użytkownik): zawiera wymagane pola obiektu użytkownik

Możliwe odpowiedzi:

- status code 201 - udało się zarejestrować
- status code 422 - błędne dane przy rejestracji

### (PATCH) Edycja użytkownika

Punkt dostępowy pozwalający na edycję profilu użytkownika.

Adres url: /api/edit

Ciało zapytania:

- payload (użytkownik): zawiera wymagane pola obiektu użytkownik

Możliwe odpowiedzi:

- status code 200 - udało się zmienić dane
- status code 400 - serwer nie mógł przetworzyć zapytania
- status code 403 - brak uprawnień do edycji użytkownika
- status code 422 - błędy w obiekcie użytkownik

### (GET) Pobieranie profilu użytkownika

Punkt dostępowy pozwalający na pobranie danych użytkownika.

Adres url: /api/user/<int:user\_id>

- user\_id - identyfikator użytkownika, którego chcemy przeglądać

Możliwe odpowiedzi:

- status code 200 - zawiera obiekt użytkownik
- status code 404 - użytkownik nie został odnaleziony

## (POST) Ocena użytkownika

Punkt dostępowy pozwalający na ocenienie użytkownika.

Adres url: /api/user/rate

Ciało zapytania:

- payload (ocena): zawiera wymagane pola obiektu ocena

Możliwe odpowiedzi:

- status code 201 - udało się zapisać ocenę
- status code 400 - serwer nie mógł przetworzyć zapytania
- status code 422 - błędne dane w obiekcie ocena

## (DELETE) Usunięcie oceny

Punkt dostępowy pozwalający na usuwanie ocen danych innym użytkownikom.

Adres url: /api/user/rating/<int:user\_id>

- user\_id - identyfikator użytkownika, któremu chcemy usunąć ocenę

Możliwe odpowiedzi:

- status code 200 - udało się usunąć ocenę
- status code 404 - nie znaleziono oceny

## (POST) Zmiana hasła

Punkt dostępowy pozwalający na zmianę hasła użytkownika.

Adres url: /api/change/

Ciało zapytania:

- payload (json object): obiekt z nowym hasłem (newPassword)

Możliwe odpowiedzi:

- status code 200 - udało się zmienić hasło
- status code 400 - serwer nie mógł przetworzyć zapytania
- status code 403 - brak uprawnień do zmiany hasła
- status code 422 - hasło nie spełnia wymagań

## (POST) Wylogowanie się

Punkt dostępowy pozwalający na wylogowanie się użytkownika.

Adres url: /api/logout/

Możliwe odpowiedzi:

- status code 200 - udało się wylogować

## (POST) Dodanie komentarza

Punkt dostępowy pozwalający na skomentowanie użytkownika.

Adres url: /chat/comment

Ciało zapytania:

- payload (komentarz): zawiera wymagane pola obiektu komentarz

Możliwe odpowiedzi:

- status code 201 - udało się zapisać komentarz
- status code 400 - serwer nie mógł przetworzyć zapytania
- status code 403 - brak uprawnień do komentowania
- status code 422 - błędy w obiekcie komentarz

## (DELETE) Usunięcie komentarza

Punkt dostępowy pozwalający na usuwanie swoich komentarzy.

Adres url: /chat/comment/<int:comment\_id>

- comment\_id - identyfikator komentarza, który chcemy usunąć

Możliwe odpowiedzi:

- status code 200 - udało się usunąć komentarz
- status code 403 - brak uprawnień do usunięcia komentarza
- status code 404 - nie znaleziono komentarza

## (GET) Pobieranie komentarzy

Punkt dostępowy pozwalający na pobranie komentarzy dotyczących użytkownika.

Adres url: /chat/getComments/<int:user\_id>

- user\_id - identyfikator użytkownika, którego komentarze chcemy pobrać

Możliwe odpowiedzi:

- status code 200 - zawiera tablicę obiektów komentarz