

Scenariusze testowe endpointów aplikacji eBook.app

1. /user/login

Metoda: POST

Opis: Logowanie użytkownika z danymi na podstawie LoginDto.

Scenariusze testowe:

- **Pozytywny:** Logowanie z poprawnym `username` i `password`.
 - **Input:**

```
{ "login": "testuser", "password": "securepassword" }
```
 - **Expected Output:**

Status 200, zwraca `userId`.
- **Negatywny:** Logowanie z niepoprawnym `username` lub `password`.
 - **Input:**

```
{ "login": "testuser", "password": "wrongpassword" }
```
 - **Expected Output:**

Status 200, zwraca `userId = -1`.

2. /user/register

Metoda: POST

Opis: Rejestracja nowego użytkownika z danymi na podstawie LoginDto.

Scenariusze testowe:

- **Pozytywny:** Rejestracja z unikalnym `username` i `password`.
 - **Input:**

```
{ "login": "newuser", "password": "securepassword123" }
```
 - **Expected Output:**

Status 200, `RegisteredDto`.
- **Negatywny:** Próba rejestracji z zajęтым `username`.
 - **Input:**

```
{ "login": "existinguser", "password": "password123" }
```
 - **Expected Output:**

Status 200, RegisteredDto, gdzie flaga isNew = false.

3. /titles/

Metoda: GET

Opis: Pobieranie listy tytułów dla użytkownika.

Scenariusze testowe:

- **Pozytywny:** Poprawny `userId`.
 - **Input:**

param: `?userId=1`
 - **Expected Output:**

Status 200, lista tytułów w formacie `TitleDto`.
- **Negatywny:** Brak istniejącego `userId`.
 - **Input:**

param: `?userId=9999`
 - **Expected Output:**

Status 200, pusta lista.

Metoda: POST

Opis: Tworzenie nowego tytułu.

Scenariusze testowe:

- **Pozytywny:** Tworzenie tytułu z poprawnymi danymi.
 - **Input:**


```
{ "userId": 1, "title": "Example Title", "author":  
  "Author Name", "year": 2024 }
```
 - **Expected Output:**

Status 200, zwraca `titleId`.
- **Pozytywny:** Tworzenie tytułu z niepoprawnym `userId`.
 - **Input:**


```
{ "userId": 999, "title": "Example Title", "author":  
  "Author Name", "year": 2024 }
```

- **Expected Output:**

Status 500, zwraca `Internal Server Error`.

Metoda: PUT

Opis: Edycja istniejącego tytułu.

Scenariusze testowe:

- **Pozytywny:** Edycja istniejącego tytułu z poprawnymi danymi.

- **Input:**

```
{ "userId": 1, "id": 1, "title": "Example Title",  
  "author": "Author Name", "year": 2024 }
```

- **Expected Output:**

Status 200, zwraca zmienione dane tytułu.

- **Pozytywny:** Edycja nieistniejącego tytułu z poprawnymi danymi.

- **Input:**

```
{ "userId": 1, "id": 999, "title": "Example Title",  
  "author": "Author Name", "year": 2024 }
```

- **Expected Output:**

Status 500, zwraca `Internal Server Error`.

Metoda: DELETE

Opis: Usuwanie tytułu

Scenariusze testowe:

- **Pozytywny:** Usunięcie istniejącego tytułu.

- **Input:**

params: `?userId=1&id=5`

- **Expected Output:**

Status 200, `true`.

- **Negatywny:** Usunięcie nieistniejącego tytułu.

- **Input:**

params: `?userId=1&id=999`

- **Expected Output:**

Status 200, `false`.

4. `/items/`

Metoda: GET

Opis: Pobieranie listy egzemplarzy dla tytułu.

Scenariusze testowe:

- **Pozytywny:** Poprawny `userId` i `titleId`.
 - **Input:**

params: `?userId=1&titleId=3`
 - **Expected Output:**

Status 200, lista egzemplarzy na podstawie `ItemDto`.
- **Negatywny:** Nieistniejący `titleId`.
 - **Input:**

params: `?userId=1&titleId=999`
 - **Expected Output:**

Status 200, pusta lista.

Metoda: POST

Opis: Tworzenie nowego egzemplarza.

Scenariusze testowe:

- **Pozytywny:** Tworzenie z poprawnymi danymi.
 - **Input:**


```
{ "userId": 1, "titleId": 3, "purchaseDate": "2024-12-01" }
```
 - **Expected Output:**

Status 200, zwraca `itemId`.
- **Negatywny:** Brak `purchaseDate`.
 - **Input:**


```
{ "userId": 1, "titleId": 3 }
```

- **Expected Output:**

Status 500, "Internal Server Error"

Metoda: PUT

Opis: Edycja istniejącego egzemplarza.

Scenariusze testowe:

- **Pozytywny:** Edycja z poprawnymi danymi.

- **Input:**

```
{ "id": 1, "userId": 3, "purchaseDate": "2024-12-01" }
```

- **Expected Output:**

Status 200, zwraca UpdatedItemDto.

- **Negatywny:** Brak purchaseDate.

- **Input:**

```
{ "userId": 1, "id": 3 }
```

- **Expected Output:**

Status 500, "Internal Server Error"

Metoda: DELETE

Opis: Usuwanie egzemplarza

Scenariusze testowe:

- **Pozytywny:** Usunięcie istniejącego egzemplarza.

- **Input:**

params: ?userId=1&id=5

- **Expected Output:**

Status 200, true.

- **Negatywny:** Usunięcie nieistniejącego egzemplarza.

- **Input:**

params: ?userId=1&id=999

- **Expected Output:**

Status 500, "Internal Server Error"

5. /rents/

Metoda: GET

Opis: Pobieranie listy wypożyczeń dla egzemplarza.

Scenariusze testowe:

- **Pozytywny:** Poprawny `userId` i `itemId`.

- **Input:**

`params: ?userId=1&itemId=10`

- **Expected Output:**

Status 200, lista wypożyczeń `RentDto`.

- **Negatywny:** Nieistniejący `itemId`.

- **Input:**

`params: ?userId=1&itemId=999`

- **Expected Output:**

Status 200, pusta lista.

Metoda: POST

Opis: Tworzenie nowego wypożyczenia.

Scenariusze testowe:

- **Pozytywny:** Poprawne dane wypożyczenia.

- **Input:**

```
{ "userId": 1, "itemId": 10, "customerName": "John Doe",  
  "rentDate": "2024-12-22", "expirationDate": "2024-12-27"  
}
```

- **Expected Output:** Status 200, zwraca `rentId`.

- **Negatywny:** Niepełne dane wypożyczenia.

- **Input:**

```
{ "userId": 1, "itemId": 10 }
```

- **Expected Output:**

Status 500, komunikat błędu.

Metoda: PUT

Opis: Edycja istniejącego wypożyczenia.

Scenariusze testowe:

- **Pozytywny:** Edycja z poprawnymi danymi.

- **Input:**

```
{ "userId": 1, "id": 1, "customerName": "John Doe",  
  "rentDate": "2024-12-22", "expirationDate": "2024-12-27"  
}
```

- **Expected Output:**

Status 200, zwraca `RentDto`.

- **Negatywny:** Brak potrzebnych danych.

- **Input:**

```
{ "userId": 1, "customerName": "John Doe" }
```

- **Expected Output:**

Status 500, `"Internal Server Error"`

Metoda: DELETE

Opis: Usuwanie wypożyczenia

Scenariusze testowe:

- **Pozytywny:** Usunięcie istniejącego wypożyczenia.

- **Input:**

params: `?userId=1&id=5`

- **Expected Output:**

Status 200, `true`.

- **Negatywny:** Usunięcie nieistniejącego wypożyczenia.

- **Input:**

params: `?userId=1&id=999`

- **Expected Output:**

Status 200, `false`.

