

Gliwice, 31.05.2022r.

Biblioteka

(dokumentacja)

Autorzy:

Katarzyna Sęk

Karol Wyligała

Krzysztof Mazur

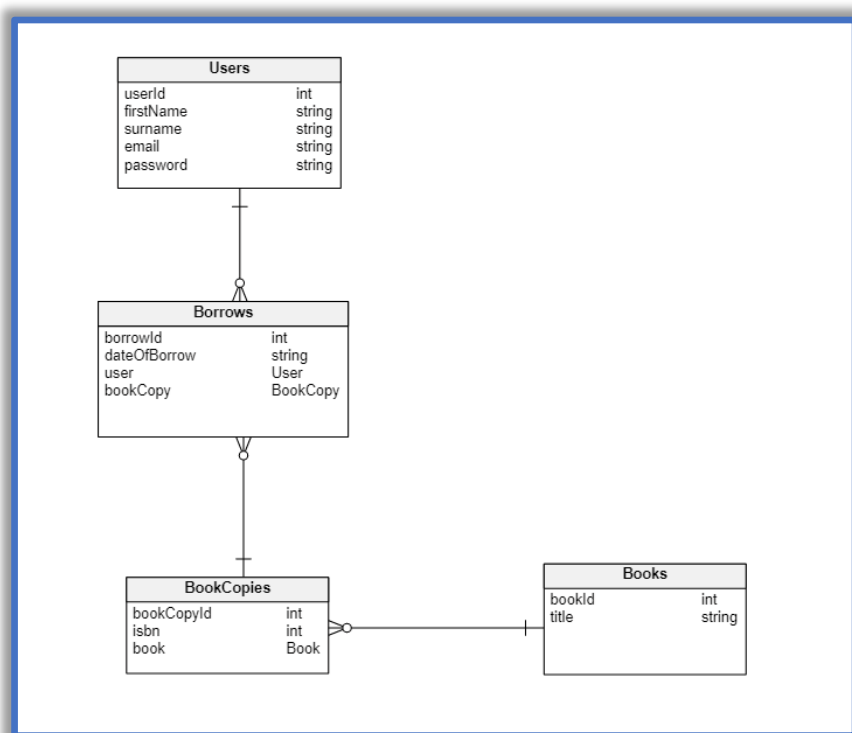
Karol Fugler

Spis treści

1. Wprowadzenie	3
2. Tabela Users	4
3. Tabela Books	6
4. Tabela BookCopies.....	9
5. Tabela Borrows.....	11
6. Podsumowanie	14

1. Wprowadzenie

Tematem naszego projektu było stworzenie i aplikacji klasy Enterprise. Do tego celu mieliśmy stworzyć bazę danych związaną tematyką z biblioteką. Po zatwierdzeniu przez Prowadzącego naszego diagramu zajęliśmy się implementacją.



Rysunek 1. diagram ERD

Do napisania aplikacji (API) wykorzystaliśmy Jave z dodatkiem Hibernate. Strona wizualna aplikacji zbudowana jest przy pomocy HTML, CSS i JavaScript.

Encje są generowane przy każdym uruchomieniu serwera, wszystkie nasze encje znajdują się w paczce *pl.project.library.entities*. Znajdują się tam również ich odpowiedniki w liczbie mnogiej, np. Book – Books, ale o tym później.

Cała logika aplikacji zawarta jest w paczce *pl.project.library.app* – tam znajduje się plik RESTowy, EJB i warstwa DAO. Z każdego punktu końcowego naszego API można skorzystać z poziomu przeglądarki. Poza tym, w dalszych

rozdziałach omówiono interakcje użytkownika z zbudowaną architekturą REST naszej aplikacji.

2. Tabela Users

Widok dostępu do tabeli Users od strony przeglądarki zaprezentowany jest na grafice poniżej.



Rysunek 2. widok obsługujący interakcje użytkownika z tabelą Users

Wszystkie możliwe operacje do przeprowadzenia na tabeli wylistowane są na czarnym pasku, wystarczy kliknąć w odpowiednią opcję.

- Dodanie użytkownika [POST]:

Adres: **`http://localhost:8080/take/library/addUser`**

Struktura XML (wysyłamy):

```
<user>
  <firstName>Krzysztof</firstName>
  <surname> Mazur </surname>
  <email> moj@mail.com </email>
</user>
```

- Pobranie użytkownika o danym id [GET]:

Adres: **`http://localhost:8080/take/library/getUser/{id}`**

Struktura XML (otrzymujemy):

```
<user>
  <email>moj@mail.com</email>
  <firstName>Krzysztof</firstName>
  <surname>Mazur</surname>
  <userId>1</userId>
</user>
```

- Pobranie wszystkich użytkowników [GET]:

Adres: **`http://localhost:8080/take/library/getUsers`**

Struktura XML (otrzymujemy):

```
<users>
  <users>
    <email>moj@mail.com</email>
    <firstName>Krzysztof</firstName>
    <surname>Mazur</surname>
    <userId>1</userId>
  </users>
  <users>
    <email>karol@drugi.pl</email>
    <firstName>Karol</firstName>
    <surname>Drugi</surname>
    <userId>10</userId>
  </users>
</users>
```

- Usunięcie użytkownika o podanym id [DELETE]:

Adres: **`http://localhost:8080/take/library/deleteUser/{id}`**

- Zaktualizowanie użytkownika [PUT]:

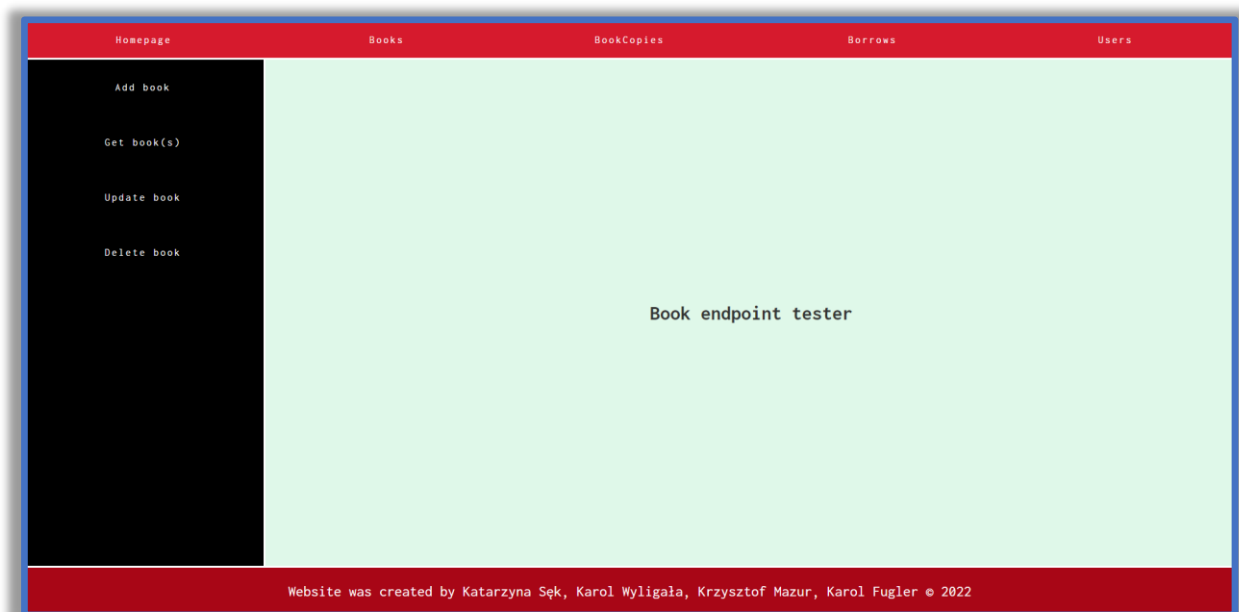
Adres: **<http://localhost:8080/take/library/updateUser>**

Struktura XML (wysyłamy):

```
<user>
  <userId>1</userId>
  <firstName>Krzysztof</firstName>
  <surname> Mazur </surname>
  <email> moj@mail.com </email>
</user>
```

3. Tabela Books

Widok dostępu do tabeli Books od strony przeglądarki zaprezentowany jest na grafice poniżej.



Wszystkie możliwe operacje do przeprowadzenia na tabeli wylistowane są na czarnym pasku, wystarczy kliknąć w odpowiednią opcję.

- Dodanie książki [POST]:

Adres: **<http://localhost:8080/take/library/addBook>**

Struktura XML (wysyłamy):

```
<book>
  <title>Harry Potter cz.1</title>
</book>
```

- Pobranie książki o danym id [GET]:

Adres: **http://localhost:8080/take/library/{id}**

Struktura XML (otrzymujemy):

```
<book>
  <bookCopies>
    <bookCopyId>4</bookCopyId>
    <isbn>1231231231231</isbn>
  </bookCopies>
  <bookCopies>
    <bookCopyId>5</bookCopyId>
    <isbn>1231231231232</isbn>
  </bookCopies>
  <bookId>1</bookId>
  <title>Harry Potter cz.1</title>
</book>
```

- Pobranie wszystkich książek [GET]:

Adres: **http://localhost:8080/take/library/getBooks**

Struktura XML (otrzymujemy):

```
<books>
  <books>
    <bookCopies>
      <bookCopyId>4</bookCopyId>
      <isbn>1231231231231</isbn>
```

```
</bookCopies>
<bookCopies>
  <bookCopyId>5</bookCopyId>
  <isbn>1231231231232</isbn>
</bookCopies>
<bookId>1</bookId>
<title>Harry Potter cz.1</title>
</books>
<books>
  <bookId>2</bookId>
  <title>Harry Potter cz.2</title>
</books>
<books>
  <bookCopies>
    <bookCopyId>6</bookCopyId>
    <isbn>1231231231239</isbn>
  </bookCopies>
  <bookId>3</bookId>
  <title>Narnia: Nieznane</title>
</books>
</books>
```

- Usunięcie książki o podanym id (oraz jej wszystkich egzemplarzy z tabeli BookCopies oraz wypożyczeń z tabeli Borrows) [DELETE]:

Adres: **<http://localhost:8080/take/library/{id}>**

- Zaktualizowanie książki [PUT]:

Adres: **<http://localhost:8080/take/library/updateBook>**

Struktura XML (wysyłamy):

```
<book>
```



```
<bookId>1</bookId>  
<title>Harry Potter cz. 3</title>  
</book>
```

4. Tabela BookCopies

Widok dostępu do tabeli BookCopies od strony przeglądarki zaprezentowany jest na grafice poniżej.



Wszystkie możliwe operacje do przeprowadzenia na tabeli wylistowane są na czarnym pasku, wystarczy kliknąć w odpowiednią opcję.

- Dodanie egzemplarza książki [POST]:

Adres: **`http://localhost:8080/take/library/addBookCopy/{id}`**

Struktura XML (wysyłamy):

```
<bookCopy>  
  <isbn>1231231231231</isbn>  
</bookCopy>
```

- Pobranie egzemplarza książki o danym id [GET]:

Adres: **<http://localhost:8080/take/library/getBookCopy/{id}>**

Struktura XML (otrzymujemy):

```
<bookCopy>
  <bookCopyId>4</bookCopyId>
  <isbn>1231231231231</isbn>
</bookCopy>
```

- Pobranie wszystkich egzemplarzy książek [GET]:

Adres: **<http://localhost:8080/take/library/getBookCopies>**

Struktura XML (otrzymujemy):

```
<bookCopies>
  <bookCopies>
    <bookCopyId>4</bookCopyId>
    <isbn>1231231231231</isbn>
  </bookCopies>
  <bookCopies>
    <bookCopyId>5</bookCopyId>
    <isbn>1231231231232</isbn>
  </bookCopies>
  <bookCopies>
    <bookCopyId>6</bookCopyId>
    <isbn>1231231231239</isbn>
  </bookCopies>
</bookCopies>
```

- Usunięcie egzemplarza książki o podanym id [DELETE]:

Adres: **<http://localhost:8080/take/library/deleteBookCopy/{id}>**

- Zaktualizowanie egzemplarza książki [PUT]:

Adres: **<http://localhost:8080/take/library/updateBookCopy>**

Struktura XML (wysyłamy):

```
<bookCopy>
  <bookCopyId>4</bookCopyId>
  <isbn>1231231231231</isbn>
</bookCopy>
```

5. Tabela Borrows

Widok dostępu do tabeli Borrows od strony przeglądarki zaprezentowany jest na grafice poniżej.



Wszystkie możliwe operacje do przeprowadzenia na tabeli wylistowane są na czarnym pasku, wystarczy kliknąć w odpowiednią opcję.

- Dodanie wypożyczenia [POST]:

Adres:

`http://localhost:8080/take/library/addBorrow/bookCopyId={bookCopyId}&userId={userId}`

Struktura XML (wysyłamy):

```
<borrow>
  <dateOfBorrow>31-05-2022</dateOfBorrow>
</borrow>
```

- Pobranie wypożyczenia o danym id [GET]:

Adres: **<http://localhost:8080/take/library/getBorrow/{id}>**

Struktura XML (otrzymujemy):

```
<borrow>
  <bookCopy>
    <bookCopyId>4</bookCopyId>
    <isbn>1231231231231</isbn>
  </bookCopy>
  <borrowId>8</borrowId>
  <dateOfBorrow>31-05-2022</dateOfBorrow>
  <user>
    <email>krzysztof@mazur.pl</email>
    <firstName>Krzysztof</firstName>
    <surname>Mazur</surname>
    <userId>7</userId>
  </user>
</borrow>
```

- Pobranie wszystkich wypożyczeń [GET]:

Adres: **<http://localhost:8080/take/library/getBorrows>**

Struktura XML (otrzymujemy):

```
<borrows>
  <borrows>
    <bookCopy>
```

```
<bookCopyId>4</bookCopyId>
<isbn>1231231231231</isbn>
</bookCopy>
<borrowId>8</borrowId>
<dateOfBorrow>31-05-2022</dateOfBorrow>
<user>
  <email>krzysztof@mazur.pl</email>
  <firstName>Krzysztof</firstName>
  <surname>Mazur</surname>
  <userId>7</userId>
</user>
</borrows>
<borrows>
  <bookCopy>
    <bookCopyId>6</bookCopyId>
    <isbn>1231231231239</isbn>
  </bookCopy>
  <borrowId>9</borrowId>
  <dateOfBorrow>31-05-2022</dateOfBorrow>
  <user>
    <email>krzysztof@mazur.pl</email>
    <firstName>Krzysztof</firstName>
    <surname>Mazur</surname>
    <userId>7</userId>
  </user>
</borrows>
</borrows>
```

- Usunięcie wypożyczenia o podanym id [DELETE]:

Adres: <http://localhost:8080/take/library/deleteBorrow/{id}>

- Zaktualizowanie wypożyczenia[PUT]:

Adres: **http://localhost:8080/take/library/updateBorrow**

Struktura XML (wysyłamy):

<bookCopy>

<borrowId>4</borrowId>

<dateOfBorrow>1231231231231</dateOfBorrow>

</bookCopy>

6. Podsumowanie

Dzięki możliwości napisania projektu w technologii, której nigdy nie używaliśmy (Hibernate), pozwoliło to nam poszerzyć światopogląd i spojrzeć na świat trochę szerzej. Poznaliśmy jego podstawowe mechanizmy, tworzenie tabel, relacji oraz wykorzystywanie jego narzędzi. Jeśli chodzi o to co zrobilibyśmy lepiej gdybyśmy mieli szanse napisać taką aplikację drugi raz, to na pewno bardziej zadbalibyśmy o widoki i bardziej precyzyjną wymianę komunikatów z serwerem.