
Instrukcja korzystania z projektu

Piotr Romaniuk

14.06.2023 – 21.06.2023

Git: [https://github.com/zkrytojojca/Conference Planner](https://github.com/zkrytojojca/Conference_Planner)

Spis treści:

- 1) Założenia przy realizacji
- 2) Baza danych
- 3) Wbudowane pomoce i dokumentacje
- 4) Usługi REST wraz z przykładowymi zapytaniami
- 5) Usługi REST a scenariusze zadania

1. Założenia przy realizacji

Zrealizowany projekt zawiera implementację serwisu służącego do obsługi konferencji IT. Podczas jego tworzenia dokonano paru założeń posiłkując się otrzymaną treścią zadania.

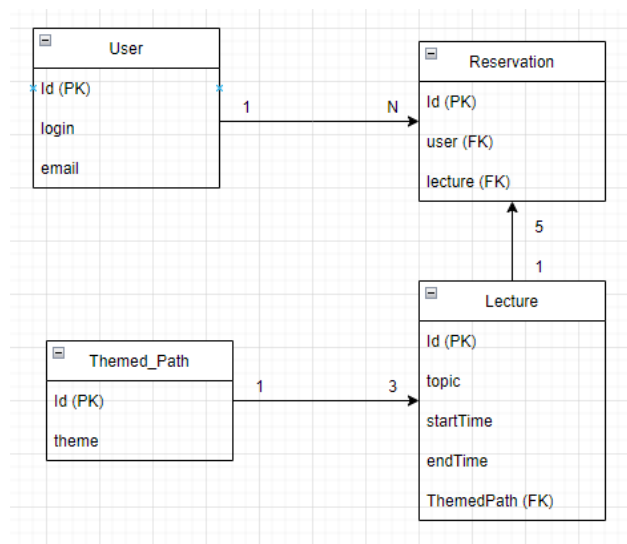
Założenia:

- 1) Kod projektu jak i komunikaty wykorzystują język angielski do komunikacji i nazewnictwa.
- 2) Skorzystano z bazy danych H2 w trybie zapisu do pliku.
- 3) Ścieżka tematyczna (ThemedPath) jest nadrzędną tematyką wykładów i składa się z kilku prelekcji.
- 4) Prelekcje (Lecture) jak i ścieżki tematyczne przechowywane są w bazie danych.
- 5) Użytkownik posiada oddzielną tabelę w bazie danych. Jego dane nie są przechowywane wyłącznie z rezerwacją prelekcji. Sprowadza się to do tego, że jego login i email są przypisane do jego „konta” a nie do konkretnej rezerwacji prelekcji.
- 6) Email użytkownika (User) oraz daty prelekcji (Lecture) nie są walidowane. Zakładam, że są one odpowiednio przekazywane od strony front-endu (aczkolwiek stosunkowo łatwo można by było dodać również walidację od strony serwisu jakby zaszła taka potrzeba).

2. Baza danych

Baza danych pracuje w trybie zapisu do pliku. Plik bazy danych znajduje się w tworzonym po uruchomieniu projektu folderze „database”. Tam również znajduje się plik o nazwie „notifications.txt” zawierający wysłane emaily z informacjami o rejestracji na prelekcje.

Schemat bazy danych wygląda następująco:



Bazę danych można również obejrzeć po uruchomieniu projektu pod adresem:
<http://localhost:8080/h2-console>

3. Wbudowane pomoce i dokumentacje

Aby podejrzeć bazę danych w projekcie lub potencjalnie wywołać ręcznie zapytania SQL, możemy skorzystać z wbudowanej konsoli bazy danych H2. Jest ona dostępna pod adresem:

<http://localhost:8080/h2-console>

Ponadto w projekcie skorzystano ze Swagger UI. Zapewnia ono automatycznie generowaną dokumentację usług REST projektu. Jest ona dostępna pod adresem:

<http://localhost:8080/swagger-ui.html>

4. Usługi REST wraz z przykładowymi zapytaniami

1) User

GET	/user
Pobiera listę wszystkich użytkowników.	
Przykładowe zapytanie: (puste)	

GET	/user/{id}
Pobiera informacje na temat użytkownika o danym id.	
Przykładowe zapytanie: (puste na id ze ścieżki)	

POST	/user
Tworzy nowego użytkownika. Zwraca komunikat na temat operacji oraz lokalizację utworzonego elementu w headers.	
Przykładowe zapytanie: Body: { "login": "login1", "email": test@gmail.com }	

PUT	/user/{id}
-----	------------

Aktualizuje użytkownika o danym id.
Przykładowe zapytanie:
Ścieżka: zawiera id
Body:
<pre>{ "login": "login2", "email": "test2@gmail.com" }</pre>

DELETE /user/{id}
Usuwa użytkownika o danym id.
Przykładowe zapytanie:
(puste na id ze ścieżki)

2) Themed Path

GET /themedPath
Pobiera listę wszystkich ścieżek tematycznych.
Przykładowe zapytanie:
(puste)

GET /themedPath/{id}
Pobiera informacje na temat ścieżki tematycznej o danym id.
Przykładowe zapytanie:
(puste na id ze ścieżki)

POST /themedPath
Tworzy nową ścieżkę tematyczną. Zwraca komunikat na temat operacji oraz lokalizację utworzonego elementu w headers.
Przykładowe zapytanie:
Body:
<pre>{ "theme": "TEST" }</pre>

PUT /themedPath/{id}
Aktualizuje ścieżkę tematyczną o danym id.
Przykładowe zapytanie:

Ścieżka: zawiera id
Body:
{
 "theme": "TEST_PUT"
}

DELETE /themedPath/{id}

Usuwa ścieżkę tematyczną o danym id.

Przykładowe zapytanie:
(puste na id ze ścieżki)

3) Lecture

GET /lecture

Pobiera listę wszystkich prelekcji.

Przykładowe zapytanie:
(puste)

GET /lecture/{id}

Pobiera informacje na temat prelekcji o danym id.

Przykładowe zapytanie:
(puste na id ze ścieżki)

POST /lecture

Tworzy nową prelekcję. Zwraca komunikat na temat operacji oraz lokalizację utworzonego elementu w headers.

Przykładowe zapytanie:

Body:

```
{  
  "lecture": {  
    "topic": "TEST",  
    "startTime": "2023-06-01T08:00:00.000+00:00",  
    "endTime": "2023-06-01T09:45:00.000+00:00"  
  },  
  "themedPathId": 1  
}
```

PUT /lecture/{id}

Aktualizuje prelekcję o danym id.

Przykładowe zapytanie:

Ścieżka: zawiera id

Body:

```
{  
  "topic": "TEST_PUT",  
  "startTime": "2024-06-01T08:00:00.000+00:00",  
  "endTime": "2024-06-01T09:45:00.000+00:00"  
}
```

DELETE /lecture/{id}

Usuwa prelekcję o danym id.

Przykładowe zapytanie:

(puste na id ze ścieżki)

4) Reservation

GET /reservation

Pobiera listę wszystkich rezerwacji.

Przykładowe zapytanie:

(puste)

GET /reservation/{id}

Pobiera informacje na temat rezerwacji o danym id.

Przykładowe zapytanie:

(puste na id ze ścieżki)

POST /reservation

Tworzy nową rezerwację. Zwraca komunikat na temat operacji oraz lokalizację utworzonego elementu w headers.

Przykładowe zapytanie:

Body:

```
{  
  "userId": 1,  
  "lectureId": 4  
}
```

POST /reservation/createUsingLoginAndEmail

Tworzy nową rezerwację na podstawie loginu i emaila a nie id użytkownika. Zwraca komunikat na temat operacji oraz lokalizację utworzonego elementu w headers.

Przykładowe zapytanie:

Body:

```
{
  "userLogin": "login1",
  "userEmail": "test@gmail.com",
  "lectureId": 4
}
```

PUT /reservation/{id}

Aktualizuje rezerwację o danym id.

Przykładowe zapytanie:

Ścieżka: zawiera id

Body:

```
{
  "user": {
    "id": 1
  },
  "lecture": {
    "id": 1
  }
}
```

DELETE /reservation/{id}

Usuwa rezerwację o danym id.

Przykładowe zapytanie:

(puste na id ze ścieżki)

GET /reservation/listUsersOfLecture

Pobiera listę wszystkich użytkowników zapisanych na daną prelekcję.

Przykładowe zapytanie:

Parametry:

Name	Value
userLogin	login2

GET /reservation/listLecturesOfUser

Pobiera listę wszystkich prelekcji zarezerwowanych przez danego użytkownika.

Przykładowe zapytanie:

Parametry:

Name	Value
lectureId	1

GET

/reservation/statistics/lectures

Pobiera zestawienie wykładów według zainteresowania.

Przykładowe zapytanie:

(puste)

GET

/reservation/statistics/themedPaths

Pobiera zestawienie ścieżek tematycznych według zainteresowania.

Przykładowe zapytanie:

(puste)

5. Usługi REST a scenariusze zadania

- 1) Użytkownik może obejrzeć plan konferencji.
GET -> /themedPath
- 2) Użytkownik po podaniu swojego loginu może obejrzeć prelekcje na które się zapisał.
GET -> /reservation/listLecturesOfUser (login jako parametr)
- 3) Jeżeli prelekcja ma jeszcze wolne miejsca, użytkownik ma możliwość dokonania rezerwacji. Podczas dokonywania rezerwacji powinien podać swój login oraz adres e-mail.
POST -> /reservation (User id + Lecture id w request body)
lub
POST -> /reservation/createUsingLoginAndEmail (User login, User email + Lecture id w request body)
- 4) Jeżeli w systemie istnieje już użytkownik z danym loginem, ale z innym adresem e-mail, system powinien zaprezentować komunikat „Podany login jest już zajęty”.
Komunikat podczas POST -> /reservation/createUsingLoginAndEmail
- 5) Poprawne dokonanie rezerwacji skutkuje wysłaniem powiadomienia użytkownikowi na podany przez niego adres.
Email jest wysyłany podczas dowolnej poprawnej rezerwacji prelekcji.

- 6) Użytkownik może anulować rezerwację.
DELETE -> /reservation/{id}
- 7) Użytkownik może zaktualizować swój adres e-mail.
PUT -> /user/{id}
- 8) System umożliwia wyświetlenie listy zarejestrowanych użytkowników wraz z ich adresami e-mail.
GET -> /user (Wszyscy użytkownicy)
lub
GET -> /reservation (Wszystkie rejestracje z id użytkowników)
- 9) Wygenerowanie zestawienia dla organizatora
 - a. zestawienie wykładów wg zainteresowania (procentowy udział uczestników w danym wykładzie)
GET -> /reservation/statistics/lectures
 - b. zestawienie ścieżek tematycznych wg zainteresowania (procentowy udział)
GET -> /reservation/statistics/themedPath
- 10) Jeśli użytkownik zapisze się w danej ścieżce na daną godzinę, to nie może już uczęszczać w tym okresie w innej ścieżce, natomiast o innej godzinie najbardziej może wybrać inną ścieżkę.
Komunikat podczas POST -> /reservation
i POST -> /reservation/creatingUsingLoginAndEmail
- 11) Każda prelekcja może pomieścić maksymalnie 5 słuchaczy.
Komunikat podczas POST -> /reservation
i POST -> /reservation/creatingUsingLoginAndEmail