

FINÁLNÍ PROJEKT

č.1



Autor: Gabriela Gajdošová
Datum: 19.10.2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	10
BUG REPORT	14

ZADÁNÍ

Cílem finálního projektu je otestovat funkčnost aplikace, která slouží k manipulaci s daty o studentech. Aplikace má rozhraní REST-API, které umožňuje vytvoření, smazání a získání dat..

Přístupové údaje:

Databáze	database: qa_demo Host: aws.connect.psdb.cloud Username: im3s0o2b4w0lpcboxk0k Password: pscale_pw_8Nc3bfP1t3ECEJm8cGPze47M egBUWQzgH7J8XsLghBi
REST-API	http://108.143.193.45:8080/api/v1/students/

Poznámky:

Nezapomeňte, že v IT se data musí někde uložit a poté získat. Proto ověřte, že data jsou správně uložena a získávána z databáze.

Nezapomeňte do testovacích scénářů uvést testovací data, očekávaný výsledek včetně těla odpovědi a stavových kódů.

TESTOVACÍ SCÉNÁŘE

Na základě uvedených testovacích scénářů jsem ověřil(a) funkčnost aplikace.

1.) Testing metody GET

Test case 1.1: Overenie, že aplikácia zobrazuje tabuľku „students“.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „GET“.
3. Klikneme na tlačítko „Send“.

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

Test case 1.2: Zobrazenie konkrétneho študenta pomocou ID.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „GET“.
3. Za http adresu napíšeme ľubovoľné existujúce ID študenta.
4. Klikneme na tlačítko „Send“.

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

Test case 1.3: Vyhľadanie študenta pomocou neexistujúceho ID.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „GET“.
3. Za http adresu napíšeme neexistujúce ID študenta.
4. Klikneme na tlačítko „Send“.

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

2.) Testing metódy POST

Test case 2.1: Vytvorenie nového študenta.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „POST“.
3. V Postmane v záložke „Body“ vložíme nasledujúci kód:

```
{  
  "firstName": "Anna",  
  "lastName": "Nováková",  
  "email": "email@gmail.com",  
  "age": 25  
}
```

4. Klikneme na tlačítko „Send“.
5. Dostaneme výstup s novým študentom a jeho unikátnym ID.
6. V programe MySQL Workbench v databáze „students“ spustíme príkaz na overenie založenia nového študenta: „SELECT * FROM student WHERE id=2220;“

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

Test case 2.2: Vytvorenie nového študenta – string v poli „age“.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „POST“.
3. V Postmane v záložke „Body“ vložíme nasledujúci kód:

```
{  
  "firstName": "Anna",  
  "lastName": "Nováková",  
  "email": "email@gmail.com",  
  "age": "dvacet"  
}
```

4. Klikneme na tlačítko „Send“.

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

Test case 2.3: Vytvorenie nového študenta – číslo v poli „firstName“.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „POST“.
3. V Postmane v záložke „Body“ vložíme nasledujúci kód:

```
{  
  "firstName": 156,  
  "lastName": "Nováková",  
  "email": "email@gmail.com",  
  "age": 25  
}
```

4. Klikneme na tlačítko „Send“.

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

Test case 2.4: Vytvorenie nového študenta – všetky polia prázdne.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „POST“.
3. V Postmane v záložke „Body“ vložíme nasledujúci kód:

```
{  
  "firstName": "",  
  "lastName": "",  
  "email": "",  
  "age":  
}
```

4. Klikneme na tlačítko „Send“.

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

Test case 2.5: Vytvorenie nového študenta – záporný vek.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „POST“.
3. V Postmane v záložke „Body“ vložíme nasledujúci kód:

```
{  
  "firstName": "Anna",  
  "lastName": "Nováková",  
  "email": "email@gmail.com ",  
  "age": -65  
}
```

4. Klikneme na tlačítko „Send“.

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

Test case 2.6: Vytvorenie nového študenta – nereálny vek.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „POST“.
3. V Postmane v záložke „Body“ vložíme nasledujúci kód:

```
{  
  "firstName": "Anna",  
  "lastName": "Nováková",  
  "email": "email@gmail.com ",  
  "age": 888888  
}
```

4. Klikneme na tlačítko „Send“.

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

Test case 2.7: Vytvorenie nového študenta so znakom v poli „age“.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „POST“.
3. V Postmane v záložke „Body“ vložíme nasledujúci kód:

```
{  
  "firstName": "Anna",  
  "lastName": "Nováková",  
  "email": "email@gmail.com ",  
  "age": "@"  
}
```

4. Klikneme na tlačítko „Send“.

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

3.) Testing metódy DELETE

Test case 3.1: Zmazanie existujúceho študenta.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „DELETE“.
3. Za http adresu napíšeme číslo ID študenta, ktorého chceme zmazať
4. Klikneme na tlačítko „Send“.
5. Výstup sa zobrazí prázdny.
6. V programe MySQL Workbench v databáze „students“ spustíme príkaz na overenie mazania študenta: „SELECT * FROM student WHERE id=2221;“

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

Test case 3.2: Zmazanie neexistujúceho študenta.

Postup:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „DELETE“.

3. Za http adresu napíšeme číslo neexistujúceho ID študenta
4. Klikneme na tlačítko „Send“.

Testovacie dáta: <http://108.143.193.45:8080/api/v1/students/>

Testovacie podmienky: OS Microsoft Windows 11 64-bit

EXEKUCE TESTŮ

Testovací scénáře jsem provedl(a), přikládám výsledky testů.

Metóda GET:

Test case 1.1: Overenie, že aplikácia zobrazuje tabuľku „students“.

Očakávaný výsledok: Program zobrazí všetky dáta z databázy „students“.

Reálny výsledok: Test prešiel – program zobrazil tabuľku „students“.

Stavový kód: 200 OK.

Test case 1.2: Zobrazenie konkrétneho študenta pomocou ID.

Očakávaný výsledok: Program zobrazí študenta, ktorý patrí pod vyhľadané ID.

Reálny výsledok: Test prešiel – program zobrazil študenta pomocou ID.

Stavový kód: 200 OK.

Test case 1.3: Vyhľadanie študenta pomocou neexistujúceho ID.

Očakávaný výsledok: Program vypíše chybu 404 a nezobrazí žiadneho študenta.

Reálny výsledok: Test neprešiel – Program nezobrazí žiadneho študenta a vypíše chybu 500.

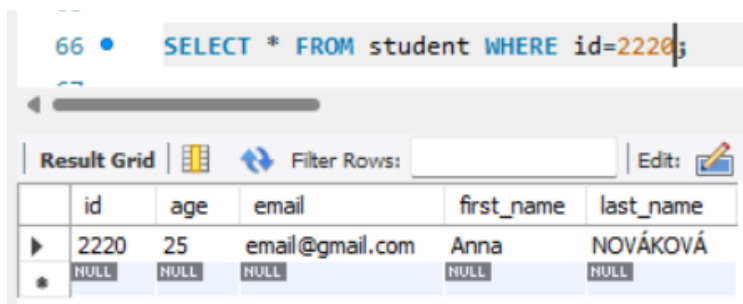
Stavový kód: 500 Internal server error.

Metóda POST:

Test case 2.1: Vytvorenie nového študenta.

Očakávaný výsledok: Program založí nového študenta a pridelí mu unikátne ID. Zmeny sa prepíšu aj do programu MySQL Workbench.

Reálny výsledok: Test prešiel – Program vytvorí nového študenta a zmena je prepísaná aj v MySQL Workbench:



The screenshot shows the MySQL Workbench interface. At the top, a SQL query is entered in the editor: `SELECT * FROM student WHERE id=2220;`. Below the editor, the 'Result Grid' tab is active, displaying the results of the query. The results are shown in a table with the following columns: id, age, email, first_name, and last_name. The first row contains the values 2220, 25, email@gmail.com, Anna, and NOVÁKOVÁ. Below this row, there are two rows of NULL values, indicating that the query returned more than one result.

	id	age	email	first_name	last_name
▶	2220	25	email@gmail.com	Anna	NOVÁKOVÁ
*	NULL	NULL	NULL	NULL	NULL

Stavový kód: 200 OK.

Test case 2.2: Vytvorenie nového študenta – string v poli „age“.

Očakávaný výsledok: Program vypíše chybu 400 – Bad request.

Reálny výsledok: Test prešiel – program vypísal chybu 400.

Stavový kód: 400 Bad request.

Test case 2.3: Vytvorenie nového študenta – číslo v poli „firstName“.

Očakávaný výsledok: Program vypíše chybu 400 – Bad request.

Reálny výsledok: Test neprešiel – program založil študenta s číslom namiesto mena.

Stavový kód: 200 OK.

Test case 2.4: Vytvorenie nového študenta – všetky polia prázdne.

Očakávaný výsledok: Program vypíše chybu 400 – Bad request.

Reálny výsledok: Test prešiel – program vypísal chybu 400.

Stavový kód: 400 Bad request.

Test case 2.5: Vytvorenie nového študenta – záporný vek.

Očakávaný výsledok: Program zobrazí chybu 400 – Bad request.

Reálny výsledok: Test neprešiel – program založil študenta so záporným vekom.

Stavový kód: 200 OK.

Test case 2.6: Vytvorenie nového študenta – nereálny vek.

Očakávaný výsledok: Program zobrazí chybu 400 – Bad request.

Reálny výsledok: Test neprešiel – program založil študenta s nereálnym vekom.

Stavový kód: 200 OK.

Test case 2.7: Vytvorenie nového študenta so znakom v poli „age“.

Očakávaný výsledok: Program zobrazí chybu 400 – Bad request.

Reálny výsledok: Test prešiel – program zobrazil chybu 400.

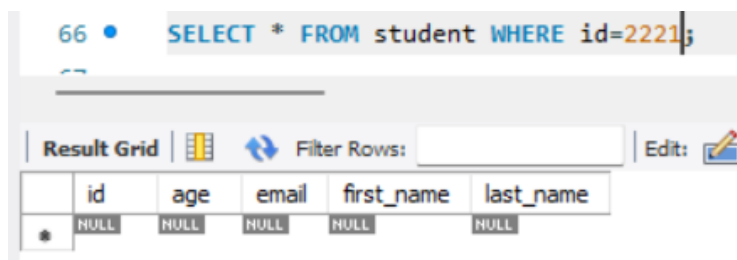
Stavový kód: 400 Bad request.

Metóda DELETE:

Test case 3.1: Zmazanie existujúceho študenta.

Očakávaný výsledok: Program zmaže vybraného študenta.

Reálny výsledok: Test prešiel – program zmazal vybraného študenta:



Stavový kód: 200 OK.

Test case 3.2: Zmazanie neexistujúceho študenta.

Očakávaný výsledok: Program vypíše chybu 404 – Not found.

Reálny výsledok: Test neprešiel – program vypísal chybu 500.

Stavový kód: 500 Internal server error.

BUG REPORT

Na základě provedených scénářů jsem objevil(a) uvedené chyby aplikace.

Test case 1.3: Vyhľadávanie pomocou neexistujúceho ID zobrazuje chybu 500.

Kroky reprodukcie:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „GET“.
3. Za http adresu napíšeme neexistujúce ID študenta.
4. Klikneme na tlačítko „Send“.

Výsledok: 500 – Internal server error.

Test case 2.3: Vytvorenie nového študenta s číslom v poli „firstName“ nezobrazí chybu 400, ale založí nový záznam.

Kroky reprodukcie:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „POST“.
3. V Postmane v záložke „Body“ vložíme nasledujúci kód:

```
{  
  "firstName": 156,  
  "lastName": "Nováková",  
  "email": "email@gmail.com",  
  "age": 25  
}
```

4. Klikneme na tlačítko „Send“.

Výsledok: Nový záznam študenta.

Test case 2.5: Program založil záznam študenta so záporným vekom.

Kroky reprodukcie:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „POST“.
3. V Postmane v záložke „Body“ vložíme nasledujúci kód:

```
{
  "firstName": "Anna",
  "lastName": "Nováková",
  "email": "email@gmail.com ",
  "age": -65
}
```

4. Klikneme na tlačítko „Send“.

Výsledok: Nový záznam študenta.

Test case 2.6: Program vytvoril záznam študenta s nereálnym vekom.

Kroky reprodukcie:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „POST“.
3. V Postmane v záložke „Body“ vložíme nasledujúci kód:

```
{
  "firstName": "Anna",
  "lastName": "Nováková",
  "email": "email@gmail.com ",
  "age": 888888
}
```

4. Klikneme na tlačítko „Send“.

Výsledok: Nový záznam študenta.

Test case 3.2: Zmazanie neexistujúceho študenta vypíše chybu 500.

Kroky reprodukcie:

1. Do poľa pre adresu URL v programe Postman vložíme http odkaz na databázu „students“.
2. Zvolíme metódu „DELETE“.
3. Za http adresu napíšeme číslo neexistujúceho ID študenta
4. Klikneme na tlačítko „Send“.

Výsledok: 500 – Internal server error.