

FINÁLNÍ PROJEKT

č.1



ENGETO

Autor: Barbora Kodlová
Datum: 29. 09. 2024

OBSAH

ZADÁNÍ.....	3
Přístupové údaje:	3
Poznámky:	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ.....	20
Testovací scénář 1: Zobrazení dat prvního existujícího studenta v databázi	20
Testovací scénář 2: Zobrazení dat posledního studenta v databázi.....	22
Testovací scénář 3: Zobrazení dat náhodného studenta v databázi	24
Testovací scénář 4: Zobrazení dat neexistujícího studenta v databázi	26
Testovací scénář 5: Smazání dat existujícího studenta	28
Testovací scénář 6: Smazání dat neexistujícího studenta	30
Testovací scénář 8: Zadání dat již existujícího studenta.....	34
Testovací scénář 9: Zadání dat již existujícího studenta s novým ID	37
Testovací scénář 10: Zadání nových dat studenta s již existujícím ID.....	40
BUG REPORT	43

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
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

1. Abstrakt: Zobrazení dat prvního existujícího studenta v databázi	
Krok	Očekávaný výsledek
1. Otevřeme program MySQL	Program je spuštěn
2. Otevřeme databázi studentů uloženou pod názvem database_students_testing	Databáze se otevře
3. Do editoru zadáme příkaz select * from student a příkaz spustíme	Zobrazí se data o všech studentech
4. Do editoru zadáme příkaz select * from student order by id asc a příkaz spustíme	Zobrazí se data o všech studentech vzestupně
5. Do editoru zadáme příkaz select * from student where id = 331 a příkaz spustíme	Zobrazí se data o prvním studentovi z databáze Id=331 Age=39 Email=john.doe@gmail.com first_name=john last_name=DOE
6. Otevřeme program Postman	Program se otevře
7. Zvolíme metodu GET	Metoda GET je nastavená
8. Zadáme URL adresu s číslem prvního existujícího studenta (id = 331): http://108.143.193.45:8080/api/v1/students/331	URL adresa je zadaná
9. Stiskneme tlačítko Send	Zobrazí se data o studentovi id = 331 Id=331

	Age=34 Email=john.doe@gmail.com first_name=john last_name=DOE
10. Zkontrolujeme zobrazené údaje o studentovi (id, firstName, lastName, email, age) z programu Postman s údaji zobrazenými v databázi studentů	Zobrazená data z programu Postman a databáze studentů jsou shodná
11. Zkontrolujeme zobrazený status kód v programu Postman	Zobrazil se status kód 200 OK

2. Abstrakt: Zobrazení dat posledního existujícího studenta v databázi	
Krok	Očekávaný výsledek
1. Otevřeme program MySQL	Program je spuštěn
2. Otevřeme databázi studentů uloženou pod názvem database_students_testing	Databáze se otevře
3. Do editoru zadáme příkaz <code>select * from student</code> a příkaz spustíme	Zobrazí se data o všech studentech
4. Do editoru zadáme příkaz <code>select * from student order by id desc</code> a příkaz spustíme	Zobrazí se data o všech studentech sestupně
5. Do editoru zadáme příkaz <code>select * from student where id = 1790</code> a příkaz spustíme	Zobrazí se data o posledním studentovi z databáze Id=1790 Age=50 Email=napis@sms.cz first_name=lenka last_name=LENKOVA
6. Otevřeme program Postman	Program se otevře
7. Zvolíme metodu GET	Metoda GET je nastavená
8. Zadáme URL adresu s číslem prvního existujícího studenta (id = 1635): <code>http://108.143.193.45:8080/api/v1/students/1790</code>	URL adresa je zadaná
9. Stiskneme tlačítko Send	Zobrazí se data o studentovi Id=1790 Age=50 Email=napis@sms.cz first_name=lenka last_name=LENKOVA

<p>10. Zkontrolujeme zobrazené údaje o studentovi (id, firstName, lastName, email, age) z programu Postman s údaji zobrazenými v databázi studentů</p>	<p>Zobrazená data z programu Postman a databáze studentů jsou shodná</p>
<p>11. Zkontrolujeme zobrazený status kód v programu Postman</p>	<p>Zobrazil se status kód 200 OK</p>

3. Abstrakt: Zobrazení dat náhodného existujícího studenta v databázi	
Krok	Očekávaný výsledek
1. Otevřeme program MySQL	Program je spuštěn
2. Otevřeme databázi studentů uloženou pod názvem database_students_testing	Databáze se otevře
3. Do editoru zadáme příkaz select * from student a příkaz spustíme	Zobrazí se data o všech studentech
4. Do editoru zadáme příkaz select * from student where id = 974 a příkaz spustíme	Zobrazí se následující data o studentovi z databáze Id=974 Age=21 Email=jsem.zde.kamo@seznam.cz first_name= last_name= KAMOS
5. Otevřeme program Postman	Program se otevře
6. Zvolíme metodu GET	Metoda GET je nastavená
7. Zadáme URL adresu s číslem posledního existujícího studenta (id = 974): http://108.143.193.45:8080/api/v1/students/974	URL adresa je zadaná
8. Stiskneme tlačítko Send	Zobrazí se data o studentovi Id=974 Age=21 Email=jsem.zde.kamo@seznam.cz first_name= last_name= KAMOS
9. Zkontrolujeme zobrazené údaje o studentovi (id, firstName, lastName, email, age) z programu Postman s údaji zobrazenými v databázi studentů	Zobrazená data z programu Postman a databáze studentů jsou shodná
10. Zkontrolujeme zobrazený status kód v programu Postman	Zobrazil se status kód 200 OK

4. Abstrakt: Zobrazení dat neexistujícího studenta

Krok	Očekávaný výsledek
1. Otevřeme program MySQL	Program je spuštěn
2. Otevřeme databázi studentů uloženou pod názvem database_students_testing	Databáze se otevře
3. Do editoru zadáme příkaz select * from student a příkaz spustíme	Zobrazí se data o všech studentech
4. Do editoru zadáme příkaz select * from student where id = 1 a příkaz spustíme	Zobrazený řádek neobsahuje žádné informace Id=0 Age=0 Email=0 first_name=0 last_name=0
5. Otevřeme program Postman	Program se otevře
6. Zvolíme metodu GET	Metoda GET je nastavená
7. Zadáme URL adresu s číslem neexistujícího studenta (id = 1): http://108.143.193.45:8080/api/v1/students/1	URL adresa je zadaná
8. Stiskneme tlačítko Send	Zobrazí se chybová hláška: "error": "Resource not found"
9. Zkontrolujeme zobrazený status kód v programu Postman	Zobrazil se status kód 404 not found

5. Abstrakt: Smazání dat existujícího studenta	
Krok	Očekávaný výsledek
1. Otevřeme program MySQL	Program je spuštěn
2. Otevřeme databázi studentů uloženou pod názvem database_students_testing	Databáze se otevře
3. Do editoru zadáme příkaz <code>select * from student</code> a příkaz spustíme	Zobrazí se data o všech studentech
4. Do editoru zadáme příkaz <code>select * from student where id = 502</code> a příkaz spustíme	Zobrazí se data o prvním studentovi z databáze Id=502 Age=19 Email=t23432@yourmail.com <u>first_name= Uzlik</u> last_name=SYN
5. Otevřeme program Postman	Program se otevře
6. Zvolíme metodu DELETE	Metoda DELETE je nastavená
7. Zadáme URL adresu s číslem prvního existujícího studenta (id = 502): <code>http://108.143.193.45:8080/api/v1/students/502</code>	URL adresa je zadaná
8. Stiskneme tlačítko Send	Zobrazí se číslo 1 a jinak žádné další záznamy
9. Zkontrolujeme zobrazený status kód v programu Postman	Zobrazil se status kód 200 OK

6. Smazání dat neexistujícího studenta	
Krok	Očekávaný výsledek
1. Otevřeme program MySQL	Program je spuštěn
2. Otevřeme databázi studentů uloženou pod názvem database_students_testing	Databáze se otevře
3. Do editoru zadáme příkaz <code>select * from student</code> a příkaz spustíme	Zobrazí se data o všech studentech
4. Do editoru zadáme příkaz <code>select * from student where id = 1</code> a spustíme příkaz	Zobrazený řádek neobsahuje žádné informace Id=0 Age=0 Email=0 first_name=0 last_name=0
5. Otevřeme program Postman	Program se otevře
6. Zvolíme metodu DELETE	Metoda DELETE je nastavená
7. Zadáme URL adresu s číslem prvního existujícího studenta (id = 1): <code>http://108.143.193.45:8080/api/v1/students/1</code>	URL adresa je zadaná
8. Stiskneme tlačítko Send	Objeví se chybová hláška "error": "Resource not found"
9. Zkontrolujeme zobrazený status kód v programu Postman	Zobrazil se status kód 404 Not Found

7. Abstrakt: Zadání dat nového neexistujícího studenta	
Krok	Očekávaný výsledek
1. Otevřeme program Postman	Program je spuštěn
2. Zvolíme metodu POST	Metoda je zvolena
3. Zadáme URL adresu http://108.143.193.45:8080/api/v1/students/	URL adresa je nastavena
4. Přepneme záložku Params na záložku Body	Záložka je přepnutá na Body
5. V levé části obrazovky zvolíme z nabídky raw	Zobrazí se textový editor připravený na zadávání údajů o studentovi
6. Záložku text přepneme do formátu JSON	JSON je nastavený
7. Do editoru zadáme údaje o studentovi: { "Id": 3, "age": 30, "email":"horak@horak.cz", "firstName":"Milan", "lastName":"Horak" }	Data jsou zadána
8. Stiskneme tlačítko Send	Zobrazí se záznamy o novém studentovi: "id": 3, "firstName":"Milan", "lastName":"HORAK", "email":"horak@horak.cz", "age":30
9. Zkontrolujeme status kód	Zobrazil se status kód 200 OK
10. Otevřeme program MySQL	Program je spuštěn
11. Otevřeme databázi studentů uloženou pod názvem database_students_testing	Databáze se otevře

<p>12. Do editoru zadáme příkaz <i>select * from student where id = 3</i> a spustíme příkaz</p>	<p>Zobrazí se data o studentovi</p> <p>Id = 3 Age = 30 Email = horak@horak.cz first_name = Milan last_name = Horak</p>
<p>13. Zkontrolujeme data o studentovi zobrazená v programu Postman s daty zobrazenými v databázi studentů</p>	<p>Data se shodují</p>

8. Abstrakt: Zadání dat již existujícího studenta

Krok	Očekávaný výsledek
1. Otevřeme program Postman	Program je spuštěn
2. Zvolíme metodu POST	Metoda je zvolena
3. Zadáme URL adresu <code>http://108.143.193.45:8080/api/v1/students/</code>	URL adresa je nastavena
4. Přepneme záložku Params na záložku Body	Záložka je přepnutá na Body
5. V levé části obrazovky zvolíme z nabídky raw	Zobrazí se textový editor připravený na zadávání údajů o studentovi
6. Záložku text přepneme do formátu JSON	JSON je nastavený
2. Otevřeme program MySQL	Program je spuštěn
3. Otevřeme databázi studentů uloženou pod názvem <code>database_students_testing</code>	Databáze se otevře
4. Do editoru zadáme příkaz <i><code>select * from student where id = 331</code></i> a spustíme příkaz	Zobrazí se data o studentovi Id = 331 Age = 34 Email = john.doe@gmail.com first_name = john last_name = DOE
7. Do editoru Postman zadáme údaje o existujícím studentovi: { "Id": 331, "firstName": "john", "lastName": "DOE", "email": "john.doe@gmail.com", "age": 34 }	Data jsou zadána

8. Stiskneme tlačítko Send	Zobrazí se hláška "Fault": "Fault user already exists"
9. Zkontrolujeme status kód	Zobrazil se status kód 409 Conflict
10. Otevřeme program MySQL	Program je spuštěn
11. Otevřeme databázi studentů uloženou pod názvem database_students_testing	Databáze se otevře
12. Do editoru zadáme příkaz <i>select * from student where id = 331</i> a spustíme příkaz	Zobrazí se data o studentovi Id = 331 Age = 34 Email = john.doe@gmail.com first_name = john last_name = DOE
13. Zkontrolujeme, že data o studentovi ID = 331 zůstala beze změny	Zobrazená data o studentovi ID = 331 se po zadání příkazu POST nezměnila

9. Abstrakt: Zadání dat již existujícího studenta s novým ID

Krok	Očekávaný výsledek
1. Otevřeme program Postman	Program je spuštěn
5. Zvolíme metodu POST	Metoda je zvolena
6. Zadáme URL adresu <code>http://108.143.193.45:8080/api/v1/students/</code>	URL adresa je nastavena
7. Přepneme záložku Params na záložku Body	Záložka je přepnutá na Body
8. V levé části obrazovky zvolíme z nabídky raw	Zobrazí se textový editor připravený na zadávání údajů o studentovi
9. Záložku text přepneme do formátu JSON	JSON je nastavený
10. Otevřeme program MySQL	Program je spuštěn
11. Otevřeme databázi studentů uloženou pod názvem database_students_testing	Databáze se otevře
12. Do editoru zadáme příkaz <i>select * from student where id = 497</i> a spustíme příkaz	Zobrazí se data o studentovi Id = 497 Age = 190 Email = zkouska.borec@yourmail.com first_name = Zkouska last_name = BOREC
13. Do editoru v programu Postman zadáme údaje o existujícím studentovi id = 497, kterému přiřadíme nové id = 5 { "Id": 5, "firstName": "Zkouška", "lastName": "BOREC", "email": "zkouska.borec@yourmail.com", "age": 190 }	Data jsou zadána

14. Stiskneme tlačítko Send	<p>Zobrazí se záznamy o novém studentovi:</p> <pre>"id": 5," "firstName": "Zkouška", "lastName": "BOREC", email":"zkouska.borec@yourmail.com ", "age": 190</pre>
15. Zkontrolujeme status kód	Zobrazil se status kód 200 OK
16. Do editoru MySQL zadáme příkaz <i>select * from student where id = 5</i> a spustíme příkaz	<p>Zobrazí se data o studentovi</p> <pre>Id = 5 Age = 190 Email = zkouska.borec@yourmail.com first_name = Zkouška last_name = BOREC</pre>
17. Zkontrolujeme, že data o studentovi ID = 5 zobrazená v databázi studentů jsou shodná se zadávanými údaji v programu Postman	Zobrazená data o studentovi ID = 5 jsou shodná

10. Abstrakt: Zadání nových dat studenta s již existujícím ID	
Krok	Očekávaný výsledek
1. Otevřeme program Postman	Program je spuštěn
2. Zvolíme metodu POST	Metoda je zvolena
3. Zadáme URL adresu http://108.143.193.45:8080/api/v1/students/	URL adresa je nastavena
4. Přepneme záložku Params na záložku Body	Záložka je přepnutá na Body
5. V levé části obrazovky zvolíme z nabídky raw	Zobrazí se textový editor připravený na zadávání údajů o studentovi
6. Záložku text přepneme do formátu JSON	JSON je nastavený
7. Otevřeme program MySQL	Program je spuštěn
8. Otevřeme databázi studentů uloženou pod názvem database_students_testing	Databáze se otevře
9. Do editoru zadáme příkaz <i>select * from student where id = 351</i> a spustíme příkaz	Zobrazí se data o studentovi Id = 351 Age = 80 Email = alfred@mail.com first_name = al last_name = HITCHCOCK
10. Do editoru v programu Postman zadáme nové údaje o studentovi id = 351 { "Id": 351, • "firstName": "Karel", • "lastName": "Karel", • "email": " test.karel@mymail.com",	Data jsou zadána

• "age": 1 }	
11. Stiskneme tlačítko Send	Zobrazí se chybová hláška např: konflikt nebo duplikace
12. Zkontrolujeme status kód	Zobrazil se status kód 409 Conflict
13. Do editoru MySQL zadáme příkaz <i>select * from student where id = 351</i> a spustíme příkaz	Zobrazí se data o studentovi Id = 351 Age = 80 Email = alfred@mail.com first_name = al last_name = HITCHCOCK
14. Zkontrolujeme, že data o studentovi ID = 351 zůstala beze změny	Zobrazená data o studentovi ID = 351 se po zadání příkazu POST nezměnila

EXEKUCE TESTŮ

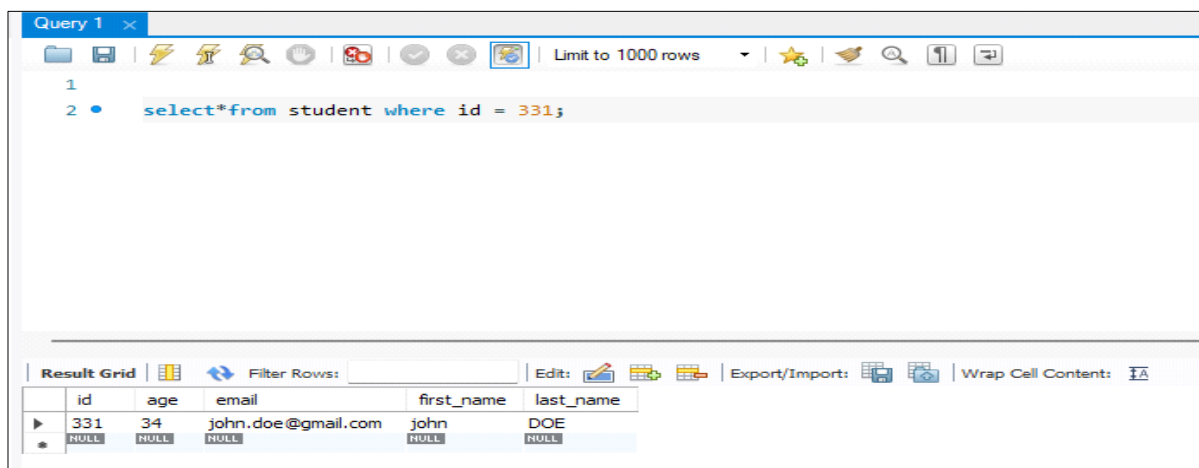
Testovací scénář 1: Zobrazení dat prvního existujícího studenta v databázi

Očekávaný výsledek: Zobrazená data z programu Postman a databáze studentů jsou shodná, zobrazí se status kód 200 OK

Skutečný výsledek: Zobrazená data z programu Postman a databáze studentů jsou shodná, zobrazil se status kód 200 OK

Poznámka: Test prošel, nebyly nalezeny žádné bugy, viz. screenshots

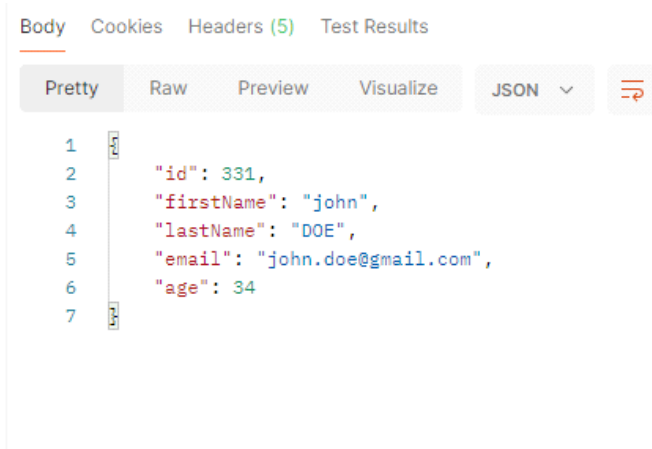
1. Data o studentovi ID = 331 zobrazená v databázi studentů



The screenshot shows a database query tool interface. At the top, there's a toolbar with various icons and a 'Limit to 1000 rows' dropdown. Below the toolbar, the SQL query is entered: `select * from student where id = 331;`. The results are displayed in a table with the following columns: id, age, email, first_name, and last_name. The first row contains the data for student ID 331: id=331, age=34, email=john.doe@gmail.com, first_name=John, last_name=DOE. Below this row, there are several rows with NULL values.

id	age	email	first_name	last_name
331	34	john.doe@gmail.com	John	DOE
NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL

2. Data o studentovi id = 331 zobrazená v programu Postman



The screenshot shows the Postman interface with the 'Body' tab selected. The response is displayed in JSON format, showing the details for student ID 331: `{ "id": 331, "firstName": "John", "lastName": "DOE", "email": "john.doe@gmail.com", "age": 34 }`.

```
1 {
2   "id": 331,
3   "firstName": "John",
4   "lastName": "DOE",
5   "email": "john.doe@gmail.com",
6   "age": 34
7 }
```

3. Status kód 200 OK

BodyCookiesHeaders (5)Test Results

PrettyRawPreviewVisualizeJSON 

12345

```
"id": 331,  
"firstName": "John",  
"lastName": "DOE",  
"email": "john.doegmail.com",
```

 Status: 200 OK

Testovací scénář 2: Zobrazení dat posledního studenta v databázi

Očekávaný výsledek: Zobrazená data z programu Postman a databáze studentů jsou shodná, zobrazí se status kód 200 OK

Skutečný výsledek: Zobrazená data z programu Postman a databáze studentů jsou shodná, zobrazil se status kód 200 OK

Poznámka: Test prošel, nebyly nalezeny žádné bugy, viz. screenshots

1. Data o studentovi id = 1790 zobrazená v databázi studentů

```
2 • select*from student where id = 1790
```

id	age	email	first_name	last_name
1790	50	napis@sms.cz	lenka	LENKOVA
NULL	NULL	NULL	NULL	NULL

2. Data o studentovi id = 1790 zobrazená v programu Postman

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1790,
3   "firstName": "lenka",
4   "lastName": "LENKOVA",
5   "email": "napis@sms.cz",
6   "age": 50
7 }
```

3. Status kód 200 OK

BodyCookiesHeaders (5)Test Results

Status: 200 OK

PrettyRawPreviewVisualize

JSON

1

2

"id": 1790,

Testovací scénář 3: Zobrazení dat náhodného studenta v databázi

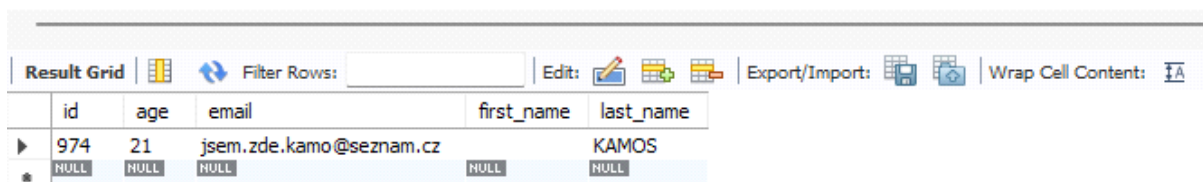
Očekávaný výsledek: Zobrazená data z programu Postman a databáze studentů jsou shodná, zobrazí se status kód 200 OK

Skutečný výsledek: Zobrazená data z programu Postman a databáze studentů jsou shodná, zobrazil se status kód 200 OK

Poznámka: Test prošel, nebyly nalezeny žádné bugy, viz. screenshots

1. Data o studentovi id = 974 zobrazená v databázi studentů

```
2 • select*from student where id = 974
```



	id	age	email	first_name	last_name
▶	974	21	jsem.zde.kamo@seznam.cz		KAMOS
*	NULL	NULL	NULL	NULL	NULL

2. Data o studentovi id = 974 zobrazená v programu Postman

Body Cookies Headers (5) Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "id": 974,
3   "firstName": "",
4   "lastName": "KAMOS",
5   "email": "jsem.zde.kamo@seznam.cz",
6   "age": 21
7 }
```


3. Status kód 200OK

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Pretty

Raw

Preview

Visualize

JSON

1

2

3

[{"id": 974,

"XXXXXXXXXX": ""

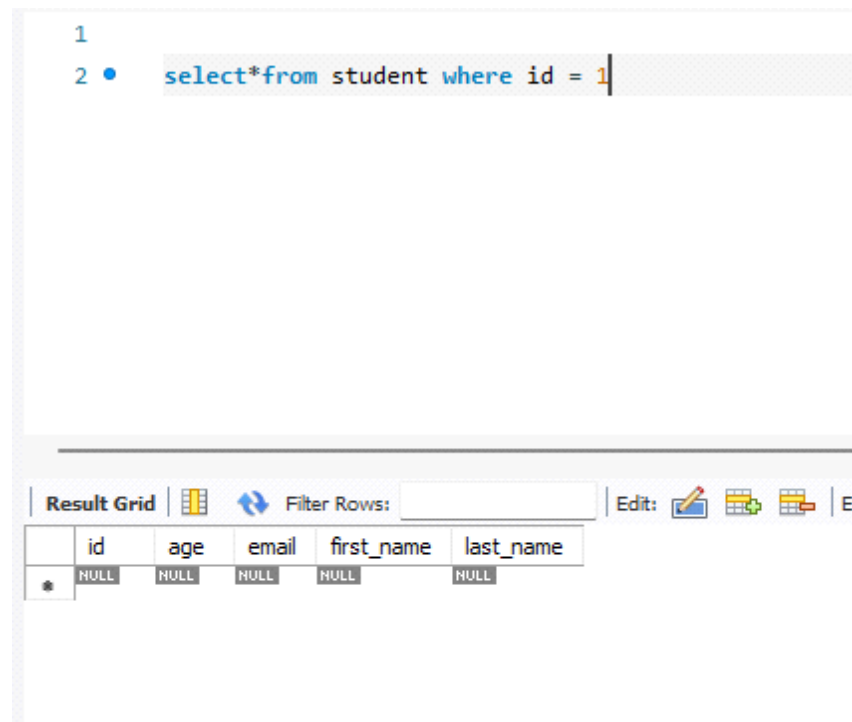
Testovací scénář 4: Zobrazení dat neexistujícího studenta v databázi

Očekávaný výsledek: V programu Postman se zobrazí se chybová hláška: "error": "Resource not found" a zobrazí se status kód 404 not found

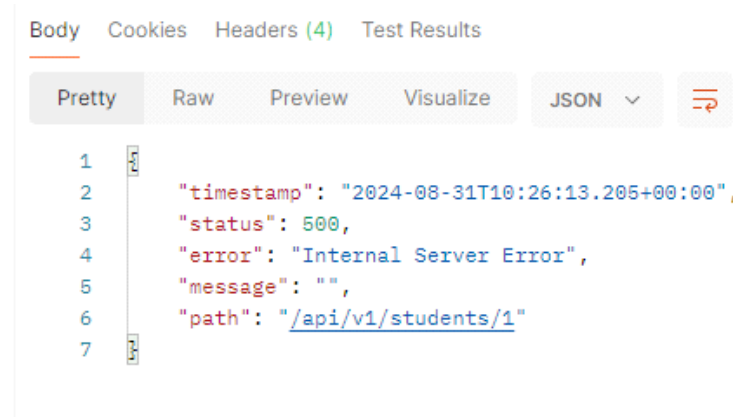
Skutečný výsledek: Po odeslání požadavku v programu Postman se objevila chybová hláška "Internal Server Error" a status kód 500 Internal Server Error, viz. screenshots

Poznámka: Bug report vytvořen pod číslem 1

1. Data o studentovi ID = 1 zobrazená v databázi studentů



2. Data o studentovi ID = 1 zobrazená v programu Postman



3. Zobrazený status kód

BodyCookiesHeaders (4)Test Results

Status: 500 Internal Server Error

PrettyRawPreviewVisualizeJSON

12

"timestamp": "2024-08-31T10:26:13.205+00:00",

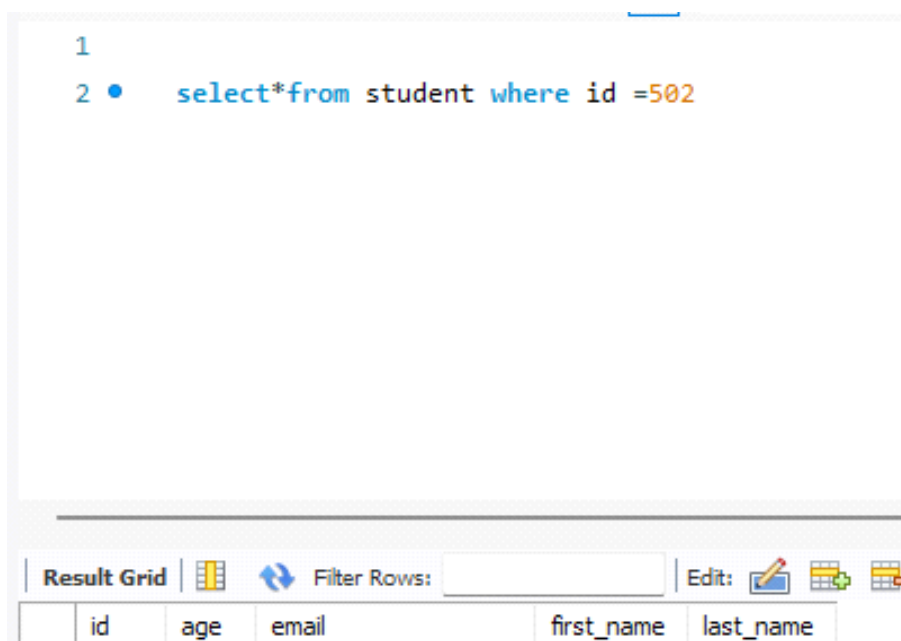
Testovací scénář 5: Smazání dat existujícího studenta

Očekávaný výsledek: Záznam o studentovi bude smazán z databáze studentů. V programu Postman se zobrazí pouze číslovka 1 jinak žádná další data. Zobrazí se status kód 200OK.

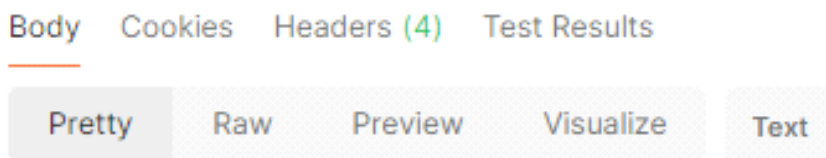
Skutečný výsledek: Po odeslání požadavku v programu Postman se objevilo číslo 1 a status kód 200 OK. V databázi studentů došlo k vymazání dat o studentovi.

Poznámka: Test prošel, nebyly nalezeny žádné buggy, viz. screenshots

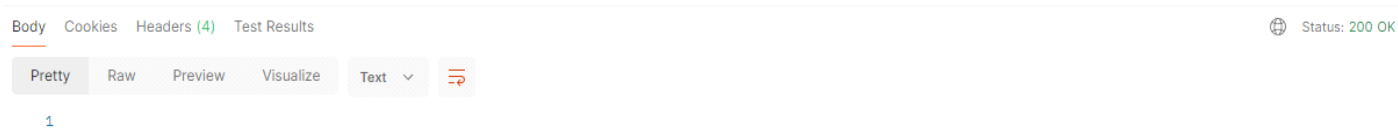
1. Data o studentovi ID = 502 zobrazená v databázi studentů



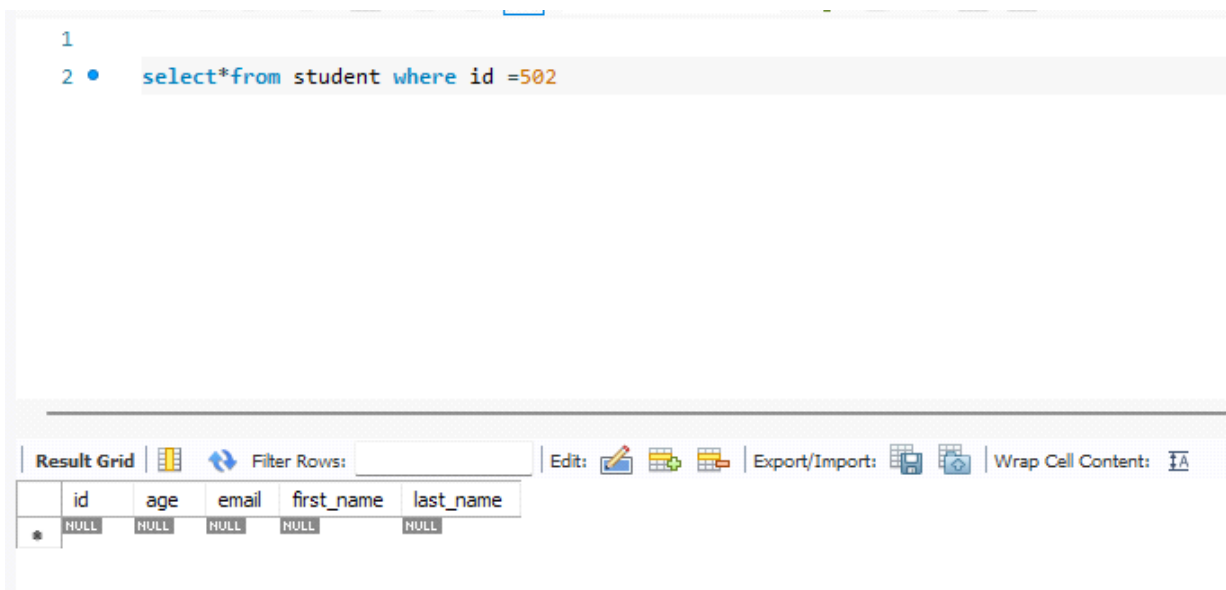
2. Data o studentovi id = 502 zobrazená v programu Postman



3. Zobrazený status kód 200OK



4. Zobrazená data v databázi studentu ID = 502



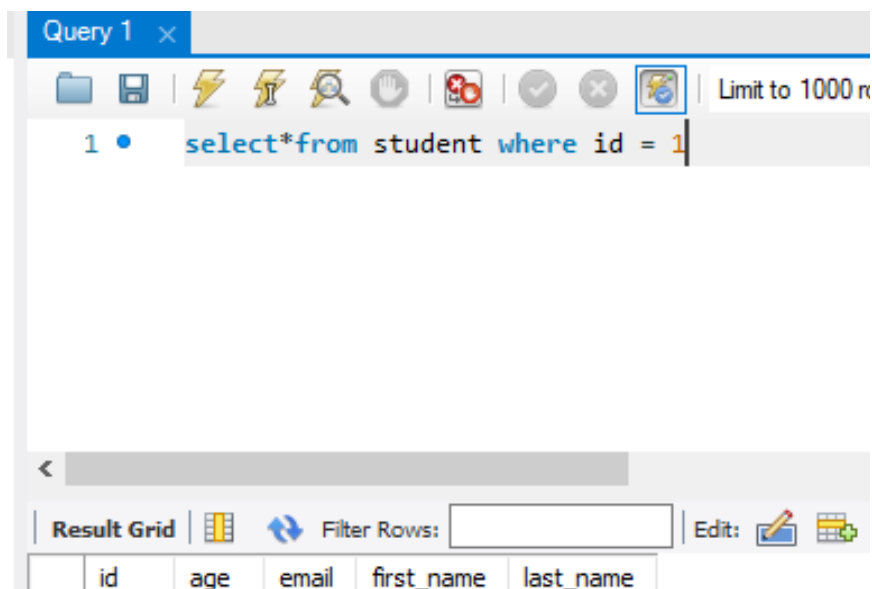
Testovací scénář 6: Smazání dat neexistujícího studenta

Očekávaný výsledek: V programu Postman se objeví chybová hláška typu "error":
"Resource not found" a zobrazí se status kód 404 Not found

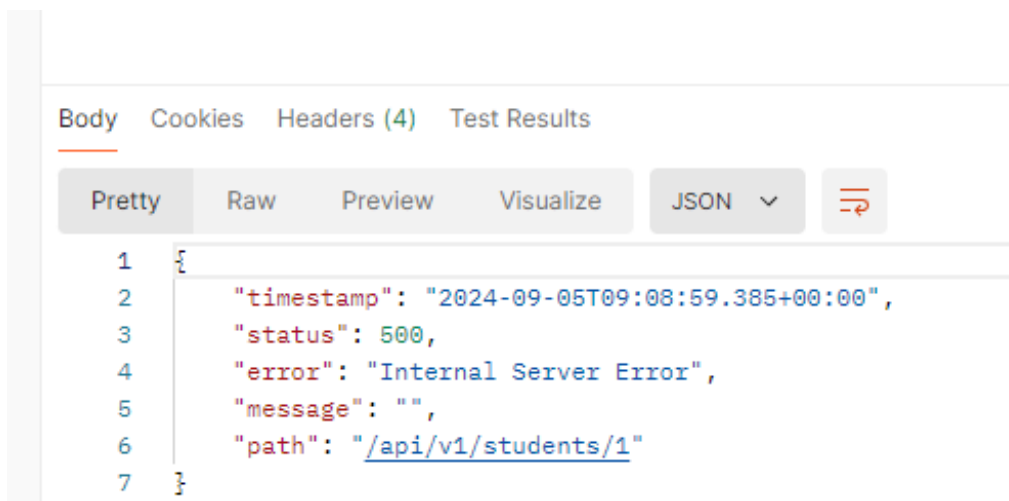
Skutečný výsledek: Po odeslání požadavku v programu Postman se objevila chybová hláška "Internal Server Error" a status kód 500 Internal Server Error, viz. screenshots

Poznámka: Bug reprot vytvořen pod číslem 2

1. Data o studentovi ID = 1 zobrazená v databázi studentů



2. Smazání dat studenta ID = 1 v programu Postman



3. Zobrazený status kód 500 Internal Server Error

Body

Cookies

Headers (4)

Test Results

Status: 500 Internal Server Error

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"timestamp": "2024-09-05T09:08:59.385+00:00",

3

"status": 500.

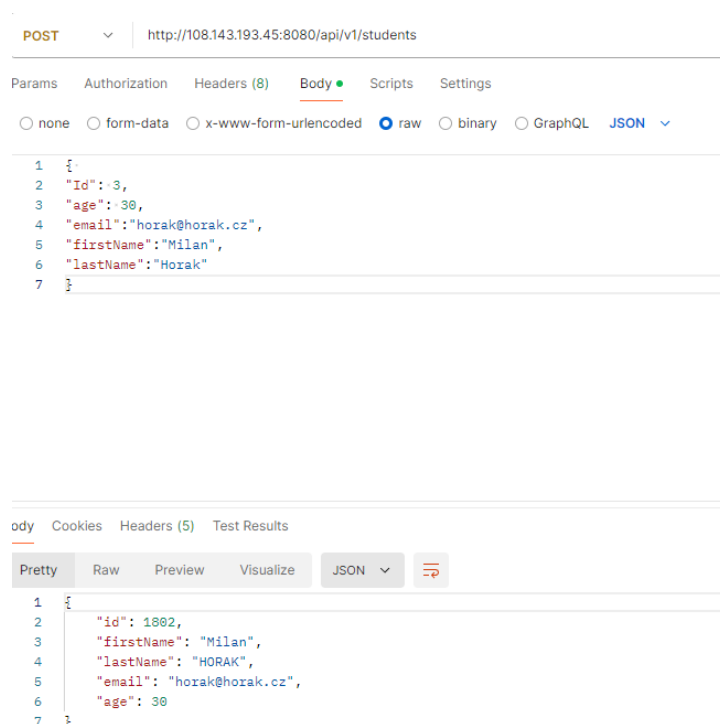
Testovací scénář 7: Zadání dat nového neexistujícího studenta

Očekávaný výsledek: V programu Postman se zobrazí data o nově zadaném studentovi ID = 3 spolu se status kódem 200 OK. Stejná data o nově zadaném studentovi ID = 3 se objeví i v databázi studentů.

Skutečný výsledek: Po odeslání požadavku v programu Postman se zobrazila nově zadaná data, ale s chybným ID = 1802. Status kód je 200 OK. V databázi studentů je daný záznam zapsán, ale pod ID = 1802. Pod požadovaným ID = 3 není nalezen žádný záznam, viz. screenshots

Poznámka: Bug report vytvořen pod číslem 3

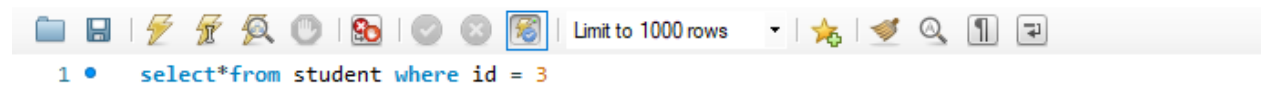
1. Zadání a zobrazení dat v programu Postman



2. Zobrazený status kód 200 OK



3. Zobrazení dat o studentovi ID = 3 v databázi



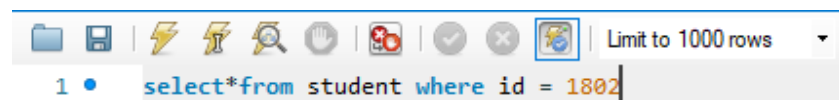
Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

	id	age	email	first_name	last_name
*	NULL	NULL	NULL	NULL	NULL

4. Zobrazení dat o studentovi pod chybným ID = 1802



Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

	id	age	email	first_name	last_name
▶	1802	30	horak@horak.cz	Milan	HORAK
*	NULL	NULL	NULL	NULL	NULL

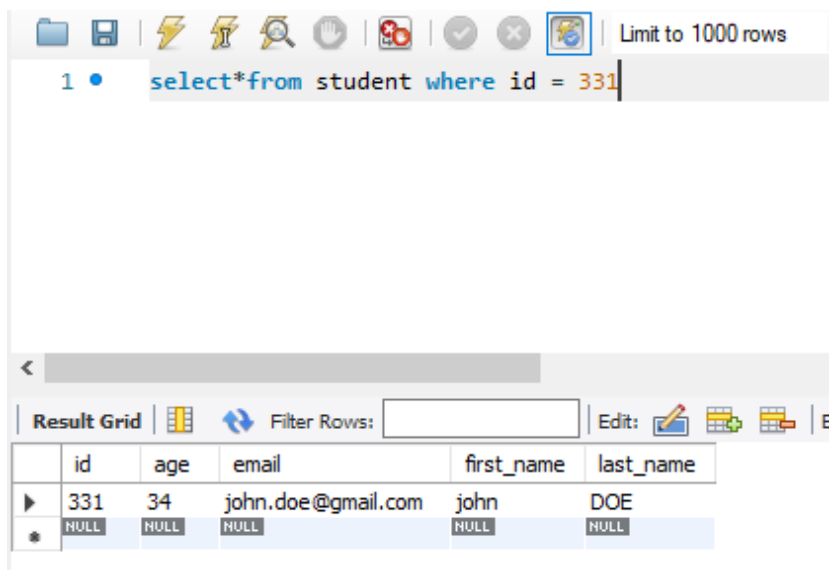
Testovací scénář 8: Zadání dat již existujícího studenta

Očekávaný výsledek: V programu Postman se zobrazí chybová hláška typu "Fault": "Fault user already exists". Zobrazí se status kód 409 Conflict. V databázi studentů nedojde k žádné změně studenta id = 331

Skutečný výsledek: Po odeslání požadavku v programu Postman se zobrazila data studenta id = 331, ale přepsána pod nové ID = 1803. Došlo tedy k přepisu dat pod náhodně vygenerované ID. V databázi studentů se aktuálně data o studentovi ID = 331 vyskytují už jen pod ID = 1803. Status kód je 200 OK, viz. screenshots

Poznámka: Bug report vytvořen pod číslem 4

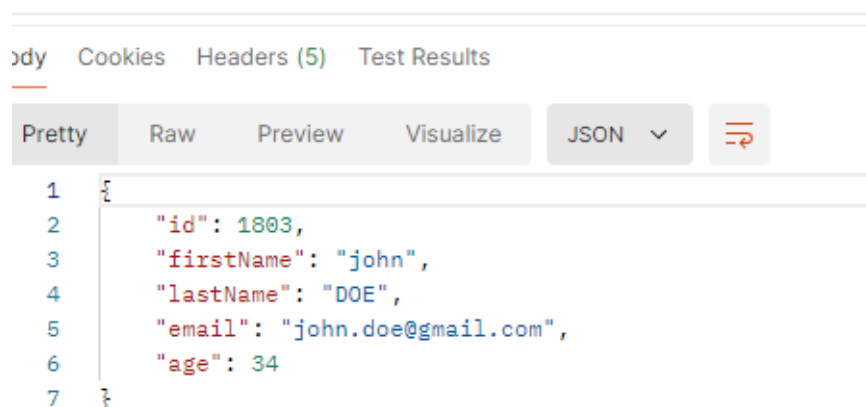
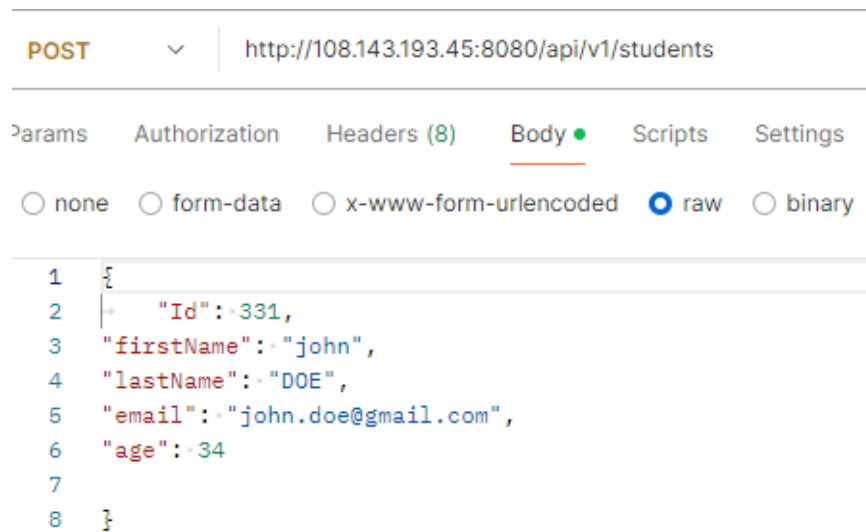
1. Zobrazení dat studenta ID = 331 v databázi studentů



The screenshot shows a database client interface. At the top, there is a toolbar with various icons. Below the toolbar, a SQL query is entered in a text field: `select * from student where id = 331`. Below the query field, there is a "Result Grid" section. It contains a table with the following data:

	id	age	email	first_name	last_name
▶	331	34	john.doe@gmail.com	john	DOE
*	NULL	NULL	NULL	NULL	NULL

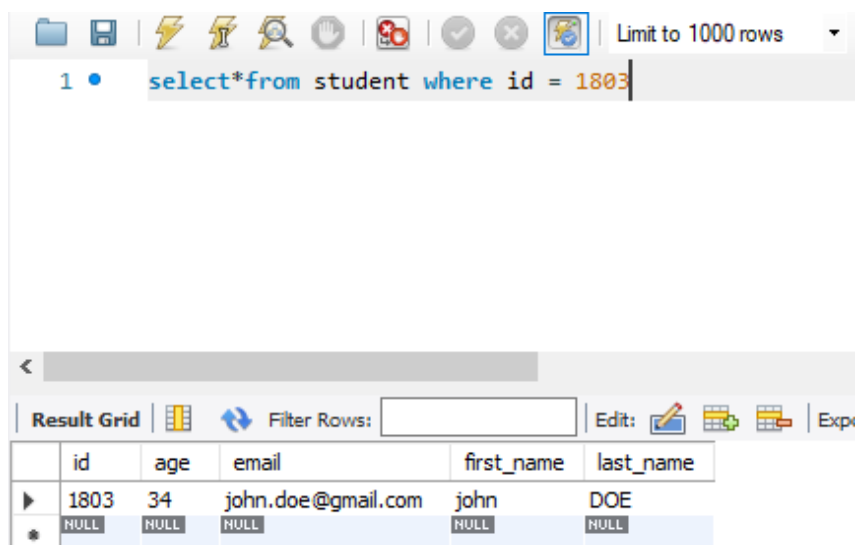
2. Zobrazení dat studenta id = 331 v programu Postman



3. Zobrazení status kódu



4. Zobrazení dat studenta ID = 1803 v databázi studentů



The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons for file operations, execution, and navigation. Below the toolbar, a SQL query is entered in a text area: `select * from student where id = 1803`. The query is preceded by a tab indicator '1 •'. To the right of the query area, there is a dropdown menu set to 'Limit to 1000 rows'. Below the query area, there is a horizontal scrollbar. Underneath the scrollbar, there is a toolbar with options for 'Result Grid', 'Filter Rows' (with an input field), 'Edit', and 'Export'. The main area displays the results of the query in a table format. The table has five columns: 'id', 'age', 'email', 'first_name', and 'last_name'. The first row of data shows the student with ID 1803, age 34, email john.doe@gmail.com, first name john, and last name DOE. Below this row, there is a row of NULL values, and a small asterisk icon is visible to the left of this row.

	id	age	email	first_name	last_name
▶	1803	34	john.doe@gmail.com	john	DOE
*	NULL	NULL	NULL	NULL	NULL

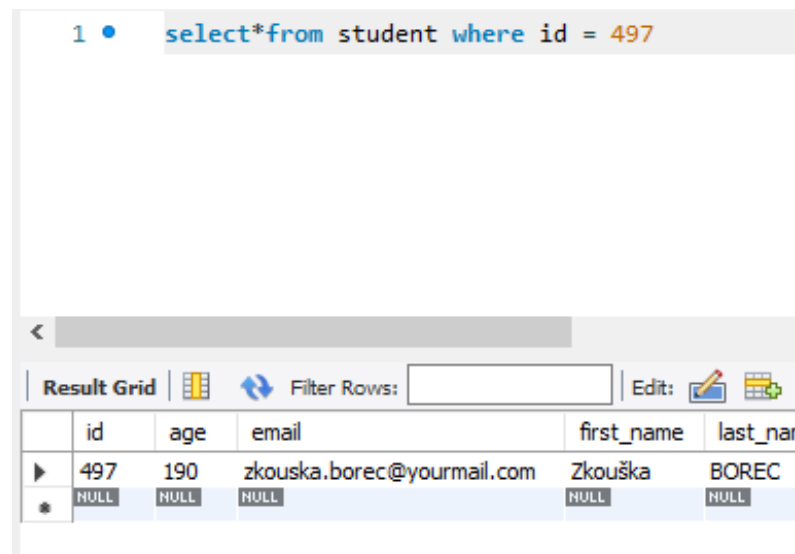
Testovací scénář 9: Zadání dat již existujícího studenta s novým ID

Očekávaný výsledek: V programu Postman se zobrazí data existujícího studenta id = 497 pod novým ID = 5. Dojde tedy k přepsání ID. V databázi studentů se zobrazí data o studentovi ID = 5, která budou shodná se zobrazenými údaji z aplikace Postman. Status kód bude 200 OK.

Skutečný výsledek: Po odeslání požadavku v programu Postman se zobrazila data studenta id = 497, ale přepsána pod nové náhodně vygenerované ID = 1804. Nedošlo tedy k přepsání na požadované ID = 5. V databázi studentů se aktuálně data o studentovi ID = 497 vyskytují už jen pod ID = 1804. Status kód je 200 OK, viz. screenshots

Poznámka: Bug report vytvořen pod číslem 5

1. Zobrazení dat studenta ID = 497 v databázi



The screenshot shows a database query interface. At the top, a SQL query is entered: `select * from student where id = 497`. Below the query, there is a toolbar with options like 'Result Grid', 'Filter Rows', and 'Edit'. The main area displays a table with the following data:

	id	age	email	first_name	last_name
▶	497	190	zkouska.borec@yourmail.com	Zkouška	BOREC
*	NULL	NULL	NULL	NULL	NULL

2. Zobrazení v programu Postman

The screenshot shows the Postman interface for a POST request. The URL is `http://108.143.193.45:8080/api/v1/students`. The request body is set to raw JSON. The JSON body is as follows:

```
1 {  
2   "Id": 5,  
3   "firstName": "Zkouška",  
4   "lastName": "BOREC",  
5   "email": "zkouska.borec@yourmail.com",  
6   "age": 190  
7 }  
8  
9 }
```

Below the request editor, the response section is visible, showing the response body in Pretty view. The response is a JSON object with the following structure:

```
1 {  
2   "id": 1804,  
3   "firstName": "Zkouška",  
4   "lastName": "BOREC",  
5   "email": "zkouska.borec@yourmail.com",  
6   "age": 190  
7 }
```

3. Status kód 200 OK






The screenshot shows the response section of Postman. The status code is 200 OK. The response body is shown in Pretty view, displaying the first part of the JSON response:

```
1 {  
2   "id": 1804,
```

4. Zobrazení dat o studentovi ID = 1804 v databázi studentů

1 • `select * from student where id = 1804`

<

Result Grid   Filter Rows: Edit:   

	id	age	email	first_name	last_name
▶	1804	190	zkouska.borec@yourmail.com	Zkouška	BOREC
*	NULL	NULL	NULL	NULL	NULL

Testovací scénář 10: Zadání nových dat studenta s již existujícím ID

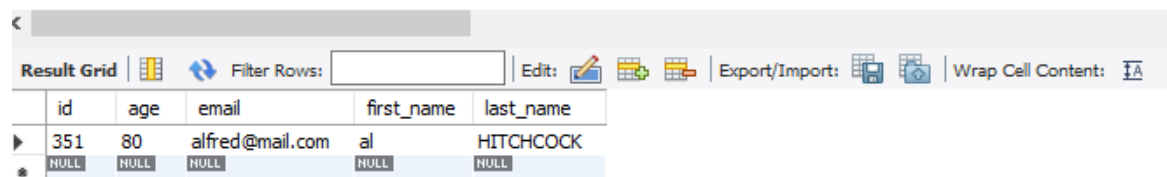
Očekávaný výsledek: V programu Postman se zobrazí chybová hláška např: konflikt nebo duplikace. Zobrazí se status kód 409 Conflict. Data o studentovi ID = 351 zůstanou beze změny a zobrazí se v databázi studentů pod tímto ID.

Skutečný výsledek: Po odeslání požadavku v programu Postman se zobrazila nově zadaná data studenta, ale přepsána pod nové náhodně vygenerované ID = 1805. Došlo tedy k přepsání ID. V databázi byl vytvořen nový záznam studenta ID = 1805. Záznam studenta ID = 351 zůstal beze změny. Status kód je 200 OK, viz. screenshots

Poznámka: Bug report vytvořen pod číslem 6

1. Zobrazení dat studenta ID = 351 v databázi

```
1 • select*from student where id = 351
```



The screenshot shows a database query result grid. The grid has a toolbar with options like 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The data is presented in a table with columns: id, age, email, first_name, and last_name. The first row contains the data for student ID 351. Below the first row, there is a row of NULL values, indicating that the query returned only one result.

	id	age	email	first_name	last_name
▶	351	80	alfred@mail.com	al	HITCHCOCK
*	NULL	NULL	NULL	NULL	NULL

2. Zadání a zobrazení dat v programu Postman


POST ▼ http://108.143.193.45:8080/api/v1/students

Params Authorization Headers (8) **Body** ● Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1  {
2    ↵
3    "Id": 351,
4    "firstName": "Karel",
5    "lastName": "Karel",
6    "email": "test.karel@mymail.com",
7    "age": 1
8  }
9
10
11 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```
1  {
2    "id": 1805,
3    "firstName": "Karel",
4    "lastName": "KAREL",
5    "email": "test.karel@mymail.com",
6    "age": 1
7  }
```

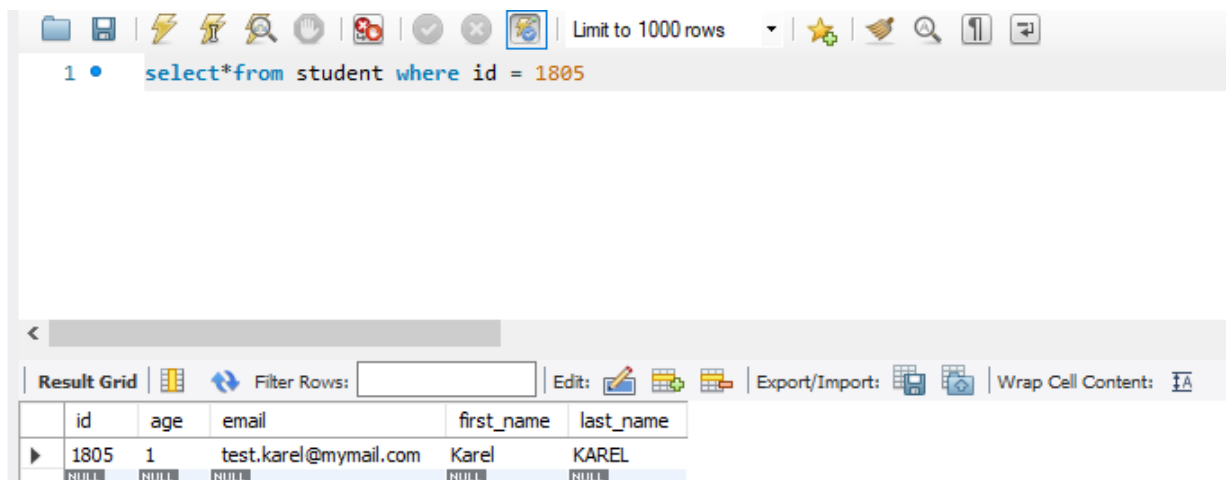
3. Status kód 200 Ok

Body Cookies Headers (5) Test Results  Status: 200 OK

Pretty Raw Preview Visualize JSON ▼ 

```
1  {
2    "id": 1805,
```

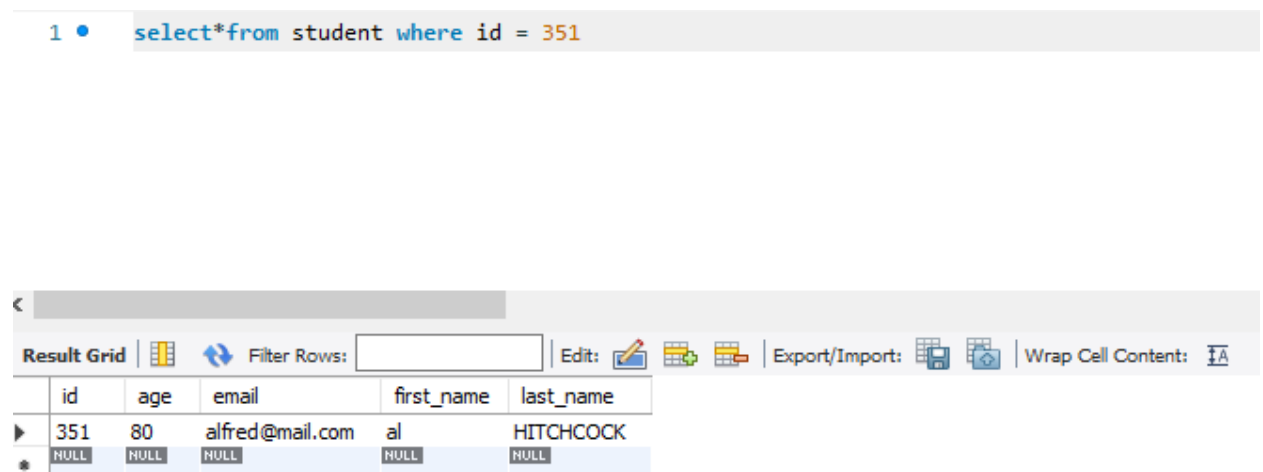
4. Zobrazení dat studenta ID = 1805 v databázi studentů



The screenshot shows a database query interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, the SQL query is entered in a text area: `1 • select * from student where id = 1805`. Below the query area, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Edit:" button, and "Export/Import:" and "Wrap Cell Content:" options. The data grid itself has columns for `id`, `age`, `email`, `first_name`, and `last_name`. The first row of data shows `id` 1805, `age` 1, `email` test.karel@mymail.com, `first_name` Karel, and `last_name` KAREL. Below this row, there are five cells, each containing the word "NULL".

	id	age	email	first_name	last_name
▶	1805	1	test.karel@mymail.com	Karel	KAREL
▲	NULL	NULL	NULL	NULL	NULL

5. Zobrazení dat studenta ID = 351



The screenshot shows a database query interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, the SQL query is entered in a text area: `1 • select * from student where id = 351`. Below the query area, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Edit:" button, and "Export/Import:" and "Wrap Cell Content:" options. The data grid itself has columns for `id`, `age`, `email`, `first_name`, and `last_name`. The first row of data shows `id` 351, `age` 80, `email` alfred@mail.com, `first_name` al, and `last_name` HITCHCOCK. Below this row, there are five cells, each containing the word "NULL".

	id	age	email	first_name	last_name
▶	351	80	alfred@mail.com	al	HITCHCOCK
*	NULL	NULL	NULL	NULL	NULL

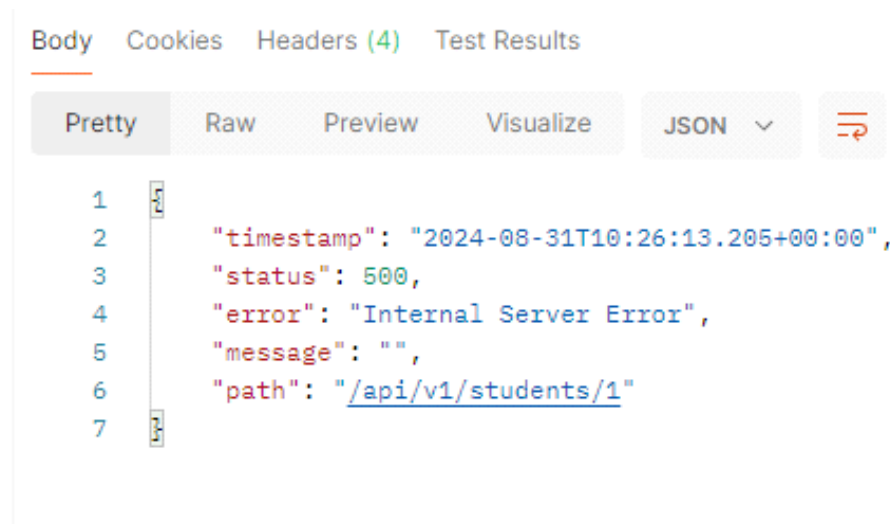
BUG REPORT

Bug report 1: Zobrazení status kódu 500 po zadání požadavku na zobrazení dat neexistujícího studenta

Kroky:

1. Otevřeme program Postman
2. Zvolíme metodu GET
3. Zadáme URL adresu: `http://108.143.193.45:8080/api/v1/students/1`
4. Stiskneme tlačítko Send

Skutečný výsledek: Po odeslání požadavku v programu Postman se objevila chybová hláška "Internal Server Error" a status kód 500 Internal Server Error, viz. screenshot



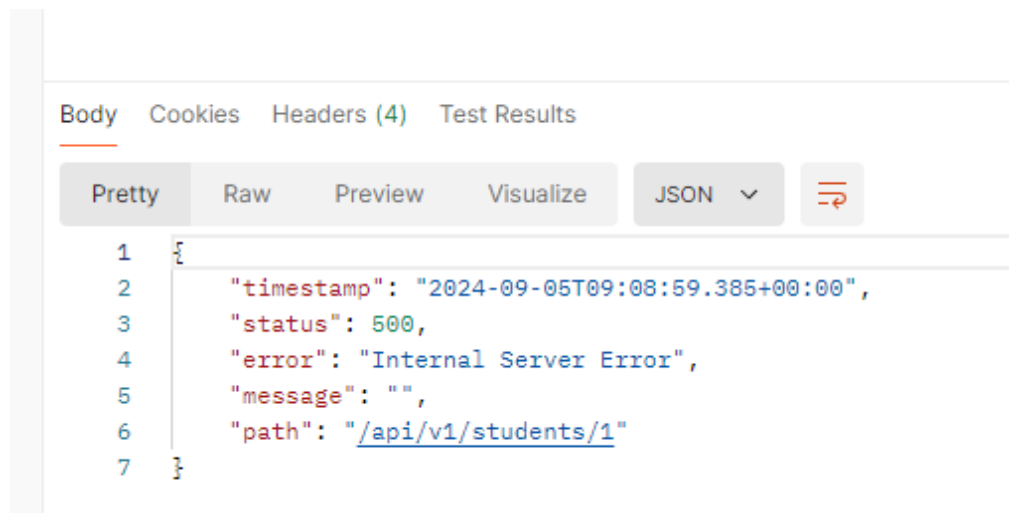
Očekávaný výsledek: Zobrazená data z programu Postman a databáze studentů jsou shodná, zobrazí se status kód 404 not found

Bug report 2: Zobrazení status kódu 500 po smazání dat neexistujícího studenta

Kroky:

1. Otevřeme program Postman
2. Zvolíme metodu DELETE
3. Zadáme URL adresu: `http://108.143.193.45:8080/api/v1/students/1`
4. Stiskneme tlačítko Send

Skutečný výsledek: Po odeslání požadavku v programu Postman se objevila chybová hláška "Internal Server Error" a status kód 500 Internal Server Error, viz. screenshot



Očekávaný výsledek: V programu Postman se objeví chybová hláška typu "error": "Resource not found" a zobrazí se status kód 404 Not found

Bug report 3: Náhodné přiřazení ID nově založenému studentovi

Kroky:

1. Otevřeme program Postman
2. Zvolíme metodu POST
3. Zadáme URL adresu: `http://108.143.193.45:8080/api/v1/students/`
4. Přepneme záložku Params na záložku Body
5. V levé části obrazovky zvolíme z nabídky raw
6. Záložku text přepneme do formátu JSON
7. Do editoru zadáme údaje o studentovi:

```
{  
  "id": 3,  
  "age": 30,  
  "email": "horak@horak.cz",  
  "firstName": "Milan",  
  "lastName": "Horak"  
}
```

8. Stiskneme tlačítko Send
9. Zkontrolujeme status kód
10. Otevřeme program MySQL
11. Otevřeme databázi studentů uloženou pod názvem `database_students_testing`
12. Do editoru zadáme příkaz `select * from student where id = 3` a spustíme příkaz
13. Zkontrolujeme data o studentovi zobrazená v programu Postman s daty zobrazenými v databázi studentů

Skutečný výsledek: Po odeslání požadavku v programu Postman se zobrazila nově zadaná data, ale s chybným ID = 1802. Status kód je 200 OK. V databázi studentů je daný záznam zapsán, ale pod ID = 1802. Pod požadovaným ID = 3 není nalezen žádný záznam, viz. screenshots

POST http://108.143.193.45:8080/api/v1/students

Params Authorization Headers (8) Body ● Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "id": 3,
3   "age": 30,
4   "email": "horak@horak.cz",
5   "firstName": "Milan",
6   "lastName": "Horak"
7 }
```

body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1802,
3   "firstName": "Milan",
4   "lastName": "HORAK",
5   "email": "horak@horak.cz",
6   "age": 30
7 }
```

Limit to 1000 rows

1 • select*from student where id = 3

Result Grid

Filter Rows: Edit: Export/Import: Wrap Cell Content:

	id	age	email	first_name	last_name
*	NULL	NULL	NULL	NULL	NULL

Očekávaný výsledek: V programu Postman se zobrazí data o nově zadaném studentovi ID = 3 spolu se status kódem 200 OK. Stejná data o nově zadaném studentovi ID = 3 se objeví i v databázi studentů.

Bug report 4: Dojde k náhodnému přepisu ID při zadávání dat existujícího studenta

Kroky:

1. Otevřeme program Postman
2. Zvolíme metodu POST
3. Zadáme URL adresu: `http://108.143.193.45:8080/api/v1/students/`
4. Přepneme záložku Params na záložku Body
5. V levé části obrazovky zvolíme z nabídky raw
6. Záložku text přepneme do formátu JSON
7. Do editoru zadáme údaje o studentovi:

```
{  
  "id": 331,  
  "firstName": "john",  
  "lastName": "DOE",  
  "email": "john.doe@gmail.com",  
  "age": 34  
}
```

8. Stiskneme tlačítko Send
9. Zkontrolujeme status kód
10. Otevřeme program MySQL
11. Otevřeme databázi studentů uloženou pod názvem `database_students_testing`
12. Do editoru zadáme příkaz `select * from student where id = 331` a spustíme příkaz
13. Zkontrolujeme, že data o studentovi ID = 331 zůstala beze změny

Skutečný výsledek: Po odeslání požadavku v programu Postman se zobrazila data studenta id = 331, ale přepsána pod nové ID = 1803. Došlo tedy k přepisu dat pod náhodně vygenerované ID. V databázi studentů se aktuálně data o studentovi ID = 331 vyskytují už jen pod ID = 1803. Status kód je 200 OK, viz. screenshots

POST



http://108.143.193.45:8080/api/v1/students

Params

Authorization

Headers (8)

Body ●

Scripts

Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary

```
1  {
2    "Id": 331,
3    "firstName": "john",
4    "lastName": "DOE",
5    "email": "john.doe@gmail.com",
6    "age": 34
7  }
8  }
```

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

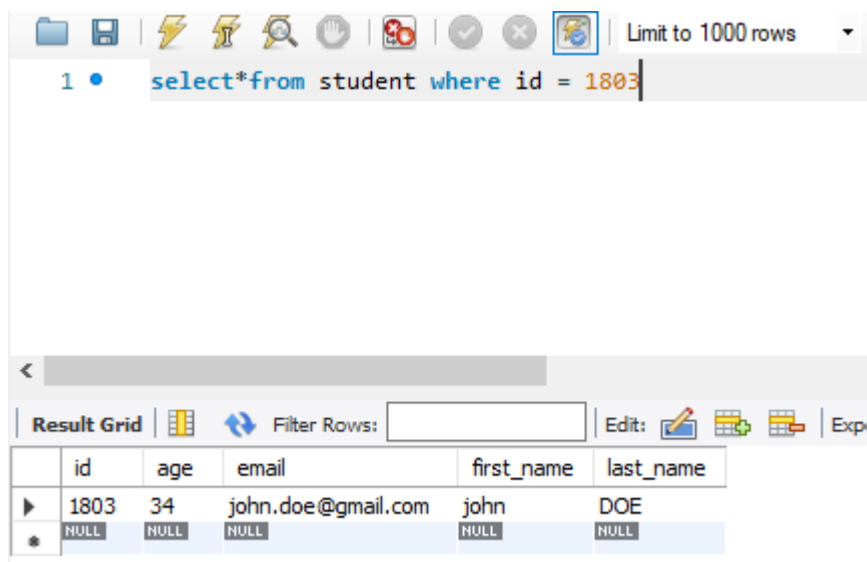
Preview

Visualize

JSON



```
1  {
2    "id": 1803,
3    "firstName": "john",
4    "lastName": "DOE",
5    "email": "john.doe@gmail.com",
6    "age": 34
7  }
```



Očekávaný výsledek: V programu Postman se zobrazí chybová hláška typu "Fault": "Fault user already exists". Zobrazí se status kód 409 Conflict. V databázi studentů nedojde k žádné změně studenta id = 331

Bug report 5: Dojde k přepisu ID existujícího studenta. ID je navíc náhodně vygenerované

Kroky:

1. Otevřeme program Postman
2. Zvolíme metodu POST
3. Zadáme URL adresu: `http://108.143.193.45:8080/api/v1/students/`
4. Přepneme záložku Params na záložku Body
5. V levé části obrazovky zvolíme z nabídky raw
6. Záložku text přepneme do formátu JSON
7. Do editoru v programu Postman zadáme údaje o existujícím studentovi `id = 497`, kterému přiřadíme nové `id = 5`

```
{  
  "id": 5,  
  "firstName": "Zkouška",  
  "lastName": "BOREC",  
  "email": "zkouska.borec@yourmail.com",  
  "age": 190  
}
```

8. Stiskneme tlačítko Send
9. Zkontrolujeme status kód
10. Otevřeme program MySQL
11. Otevřeme databázi studentů uloženou pod názvem `database_students_testing`
12. Do editoru MySQL zadáme příkaz `select * from student where id = 5` a spustíme příkaz
13. Zkontrolujeme, že data o studentovi `ID = 5` zobrazená v databázi studentů jsou shodná se zadávanými údaji v programu Postman

Očekávaný výsledek: V programu Postman se zobrazí data existujícího studenta `id = 497` pod novým `ID = 5`. Dojde tedy k přepsání ID. V databázi studentů se zobrazí data o studentovi `ID = 5`, která budou shodná se zobrazenými údaji z aplikace Postman. Status kód bude 200 OK.

Skutečný výsledek: Po odeslání požadavku v programu Postman se zobrazila data studenta `id = 497`, ale přepsána pod nové náhodně vygenerované `ID = 1804`. Nedošlo tedy k

přepsání na požadované ID = 5. V databázi studentů se aktuálně data o studentovi ID = 497 vyskytují už jen pod ID = 1804. Status kód je 200 OK, viz. screenshots

POST

▼

http://108.143.193.45:8080/api/v1/students

ParamsAuthorizationHeaders (8)Body ●ScriptsSettings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON ▼




```
1  {
2    →  "Id": 5,
3    →  "firstName": "Zkouška",
4    →  "lastName": "BOREC",
5    →  "email": "zkouska.borec@yourmail.com",
6    →  "age": 190
7
8
9  }
```

BodyCookiesHeaders (5)Test Results

PrettyRawPreviewVisualizeJSON ▼

```
1  {
2    "id": 1804,
3    "firstName": "Zkouška",
4    "lastName": "BOREC",
5    "email": "zkouska.borec@yourmail.com",
6    "age": 190
7  }
```

```
1 • select*from student where id = 1804
```

<					
Result Grid					
Filter Rows: <input type="text"/>					
Edit:   					
	id	age	email	first_name	last_name
▶	1804	190	zkouska.borec@yourmail.com	Zkouška	BOREC
*	NULL	NULL	NULL	NULL	NULL

Bug report 6: Při zadávání nových dat o studentovi pod již existující ID dojde k náhodnému přepisu ID

Kroky:

1. Otevřeme program Postman
2. Zvolíme metodu POST
3. Zadáme URL adresu: `http://108.143.193.45:8080/api/v1/students/`
4. Přepneme záložku Params na záložku Body
5. V levé části obrazovky zvolíme z nabídky raw
6. Záložku text přepneme do formátu JSON
7. Do editoru v programu Postman zadáme nové údaje o studentovi `id = 351`

```
{  
    "id": 351,  
    "firstName": "Karel",  
    "lastName": "Karel",  
    "email": " test.karel@mymail.com",  
    "age": 1  
}
```

8. Stiskneme tlačítko Send
9. Zkontrolujeme status kód
10. Otevřeme program MySQL
11. Otevřeme databázi studentů uloženou pod názvem `database_students_testing`
12. Do editoru MySQL zadáme příkaz `select * from student where id = 351` a spustíme příkaz
13. Zkontrolujeme, že data o studentovi ID = 351 zůstala beze změny

Očekávaný výsledek: V programu Postman se zobrazí chybová hláška např: konflikt nebo duplikace. Zobrazí se status kód 409 Conflict. Data o studentovi ID = 351 zůstanou beze změny a zobrazí se v databázi studentů pod tímto ID.

Skutečný výsledek: Po odeslání požadavku v programu Postman se zobrazila nově zadaná data studenta, ale přepsána pod nové náhodně vygenerované ID = 1805. Došlo tedy k přepsání ID. V databázi byl vytvořen nový záznam studenta ID = 1805. Záznam studenta ID = 351 zůstal beze změny. Status kód je 200 OK, viz. screenshots

The screenshot displays the Postman interface for a POST request to the endpoint `http://108.143.193.45:8080/api/v1/students`. The request body is a JSON object with the following fields: `Id` (351), `firstName` (Karel), `lastName` (Karel), `email` (test.karel@mymail.com), and `age` (1). The response body, shown in the 'Body' tab, is a JSON object with the following fields: `id` (1805), `firstName` (Karel), `lastName` (KAREL), `email` (test.karel@mymail.com), and `age` (1). The response is formatted as 'Pretty' JSON.

```
1 {
2   "Id": 351,
3   "firstName": "Karel",
4   "lastName": "Karel",
5   "email": "test.karel@mymail.com",
6   "age": 1
7 }
```

```
1 {
2   "id": 1805,
3   "firstName": "Karel",
4   "lastName": "KAREL",
5   "email": "test.karel@mymail.com",
6   "age": 1
7 }
```

The screenshot shows the SQL editor with the query `select * from student where id = 1805`. Below the editor, the 'Result Grid' is displayed, showing the following data:

	id	age	email	first_name	last_name
▶	1805	1	test.karel@mymail.com	Karel	KAREL
*	NULL	NULL	NULL	NULL	NULL