

FINÁLNÍ PROJEKT

č.1



Autor: Tereza Bocánová
Datum: 18.4.2024

Obsah

Seznam obrázků	3
ZADÁNÍ.....	4
Přístupové údaje:	4
Poznámky:	4
TESTOVACÍ SCÉNÁŘE	5
GET	5
Zaměřeno na studenta s ID 350.....	5
Zaměřeno na studenta s ID 400.....	6
Zaměřeno na studenta s ID 427.....	8
Zaměřeno na studenta s ID 999 – neexistující student.....	9
Zaměřeno na studenta „Novák“ – vícero stejných jmen	11
POST	13
Vytvořen nový student Tereza Bocánová	13
Vytvořen nový student Anna-Maria Luisa Řehořčíčková	14
Vytvořen nový student jana ze 3.patra 123+-*/@	16
Vytvořen nový student Mgr. Ing. Jan III. Müller L' de Fořt PhD.	17
DELETE.....	18
Smazán student s ID 591	18
Smazán student s ID 186	18
Smazán student s ID 123456 – neexistující student	19
ZÁVĚR.....	21

Seznam obrázků

Obrázek 1: Postman: GET, ID studenta 350.....	5
Obrázek 2: MSW: GET, ID studenta 350	6
Obrázek 3: Postman: GET, ID studenta 400.....	7
Obrázek 4: MSW: GET, ID studenta 400	7
Obrázek 5: Postman: GET, ID studenta 427.....	8
Obrázek 6: MSW: GET, ID studenta 427	9
Obrázek 7: Postman: GET, ID studenta 999 – neexistující student.....	10
Obrázek 8: MSW: GET, ID studenta 999 - neexistující student	10
Obrázek 9: Postman: GET, Novák - vícero stejných jmen	11
Obrázek 10: MSW: GET, Novák - vícero stejných jmen	12
Obrázek 11: Postman: POST, Tereza Bocánová	13
Obrázek 12: MSW: POST, ID studenta 590	14
Obrázek 13: Postman: POST, Anna-Maria Luisa Řehořčíčková	15
Obrázek 14: MSW: POST, Anna-Maria Luisa Řehořčíčková	15
Obrázek 15: Postman: POST, jana ze 3.patra 123+-*/*@	16
Obrázek 16: Postman: POST, Mgr. Ing. Jan III. Müller L' de Foňt PhD.	17
Obrázek 17: MSW: DELETE student s ID 591	18
Obrázek 18: MSW: DELETE student s ID 186	19
Obrázek 19: Postman: DELETE, student s ID 123456 - neexistující student.....	20
Obrázek 20: MSW: DELETE, student s ID 123456 - neexistující student	20

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: 68q0llcbuyijdt5mzq2 Password: pscale_pw_zxfenAXfSYx9loe9PCS1snuGKrFzw2J84BMIVZI96o
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

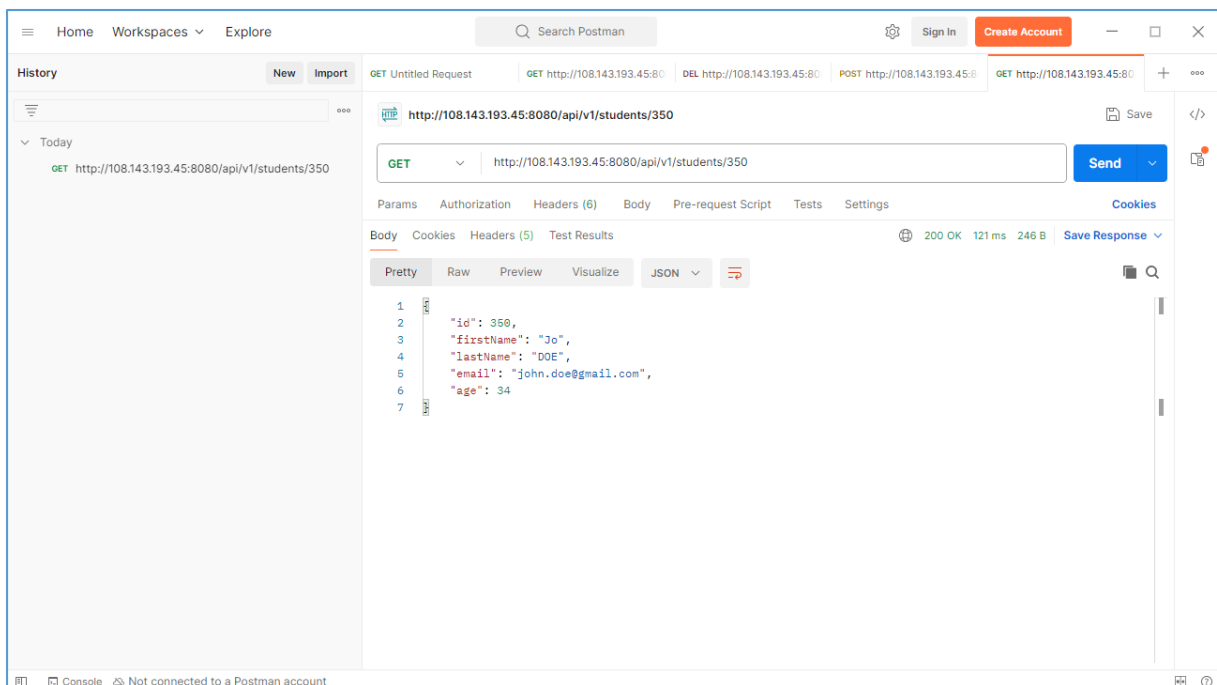
- Testováno MySQLWorkbench 8.0
 - Nainstalováno dle pokynů, spárováno dle uvedených přístupových údajů. Vše bez problému.
- Testováno v Postman
 - Nainstalováno dle pokynů. Vše bez problémů.

GET

Nejdříve jsem testovala funkci „Get“, která slouží k vyhledání dat. Pro svoje testování jsem si náhodně vybrala studenty s ID 350, 400 a 427. Zaměřila jsem se i na neexistujícího studenta s ID 999 a zkusila jsem také vyhledávat podle příjmení „Novák“.

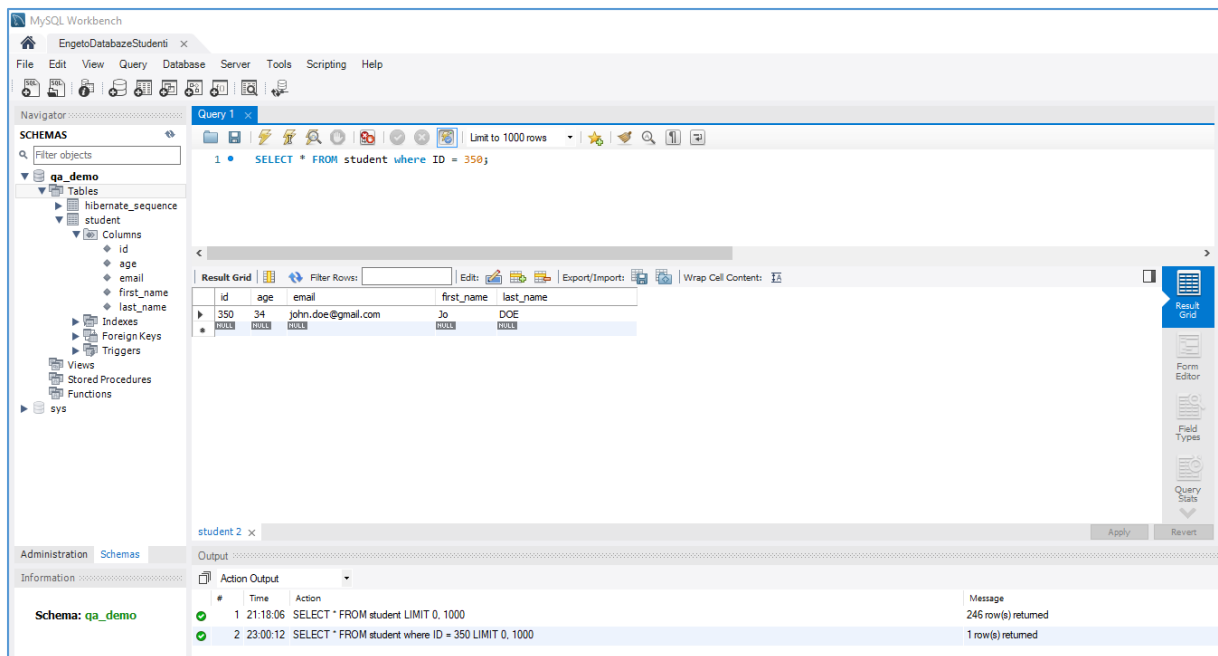
Zaměřeno na studenta s ID 350

1. Otevřela jsem si aplikaci Postman
 - a. „New“ → „http“ → roletka „GET“ → odkaz: <http://108.143.193.45:8080/api/v1/students/350> → „Send“.



Obrázek 1: Postman: GET, ID studenta 350

2. Otevřela jsem aplikaci MySQLWorkbench
 - a. SQL příkaz: `SELECT * FROM student WHERE id = 350;`

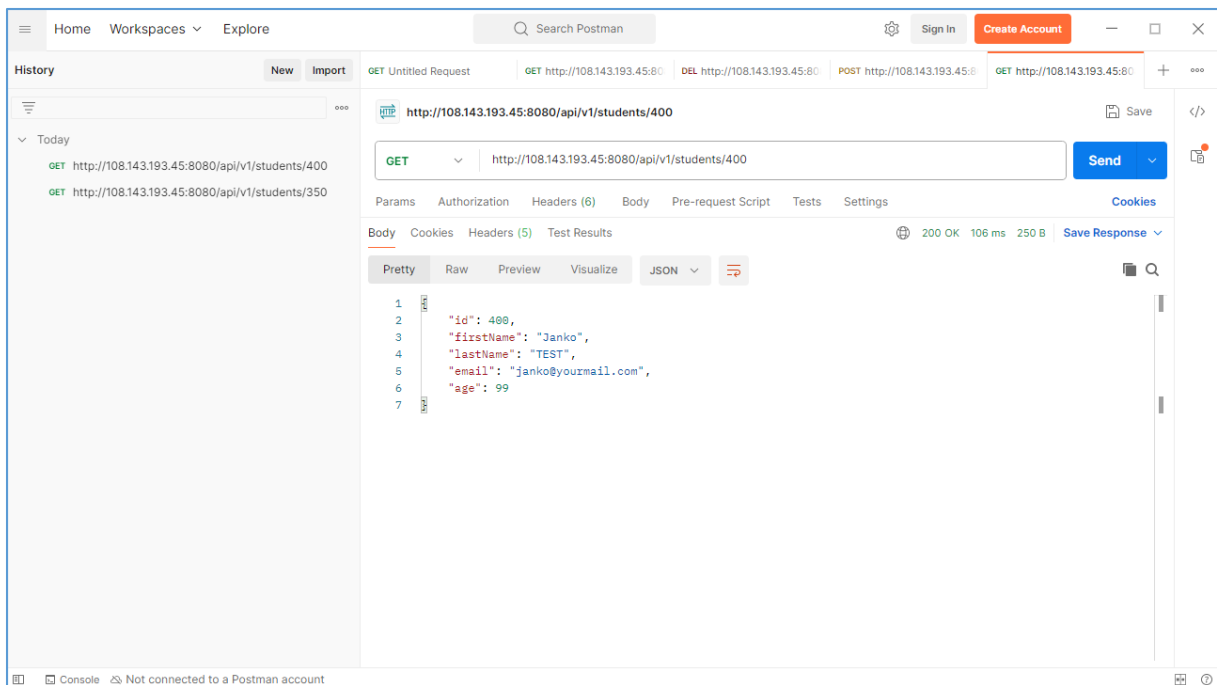


Obrázek 2: MSW: GET, ID studenta 350

- Očekávání:
 - o Status „200 OK“ = Potvrzeno.
 - o V databázi je vyhledán student s ID 350, kde jeho první jméno je Jo, příjmení DOE, věk 34, email john.doe@gmail.com
- Reálný výsledek: SOUHLASÍ, Chyba se neobjevila.

Zaměřeno na studenta s ID 400

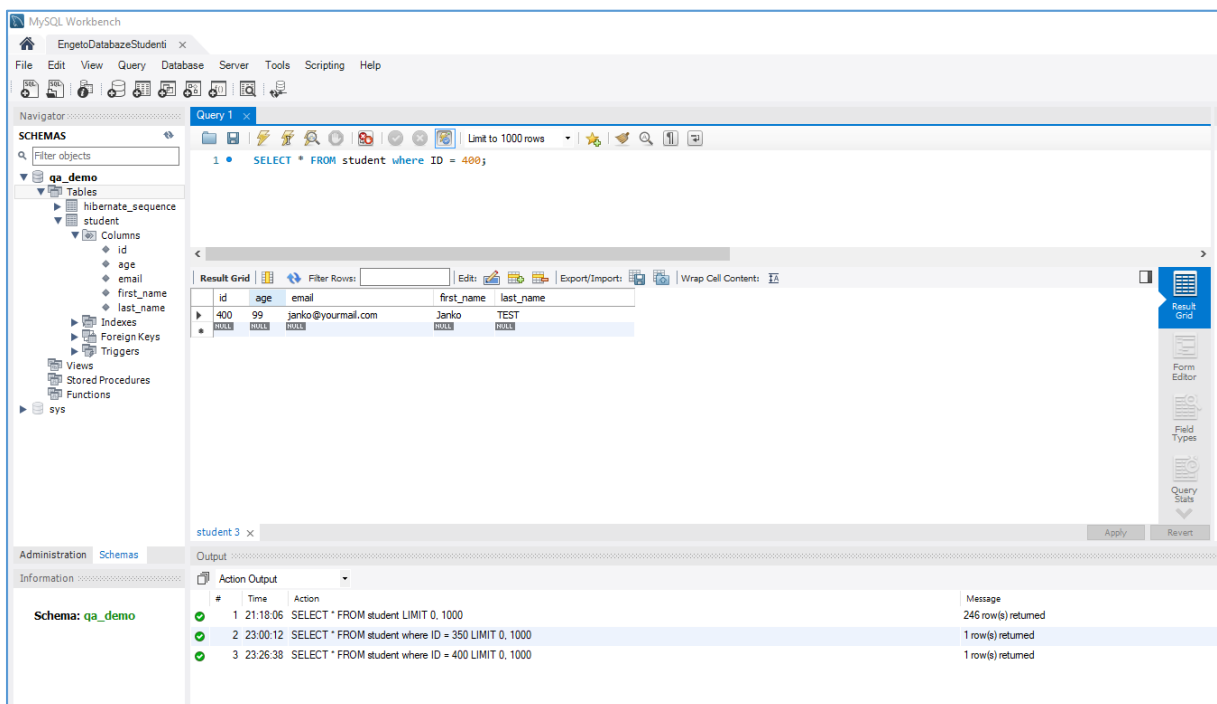
1. Otevřela jsem si aplikaci Postman
 - a. „New“ → „http“ → roletka „GET“ → odkaz: <http://108.143.193.45:8080/api/v1/students/400> → „Send“.



Obrázek 3: Postman: GET, ID studenta 400

2. Otevřela jsem aplikaci MySQLWorkbench

- a. SQL příkaz: `SELECT * FROM student WHERE id = 400;`



Obrázek 4: MSW: GET, ID studenta 400

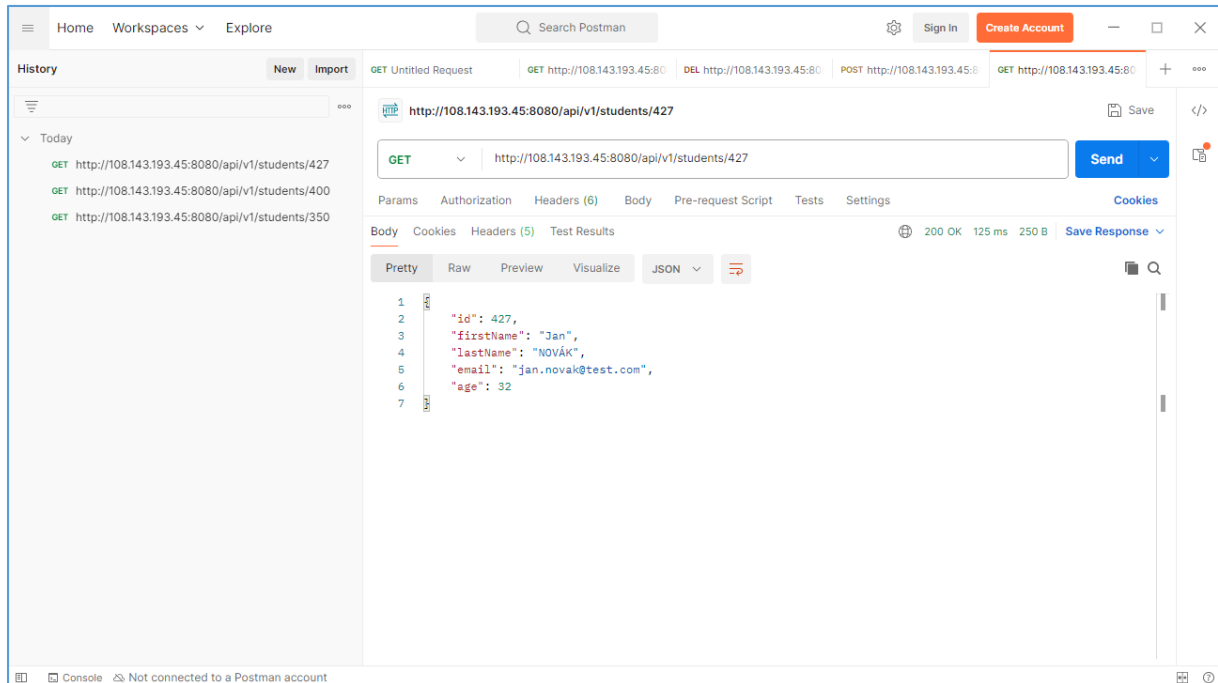
- Očekávání:

- Status „200 OK“ – Potvrzeno.
- V databázi je vyhledán student s ID 400, kde jeho první jméno je Janko, příjmení TEST, věk 99, email janko@yourmail.com

- Reálný výsledek: SOUHLASÍ, Chyba se neobjevila.

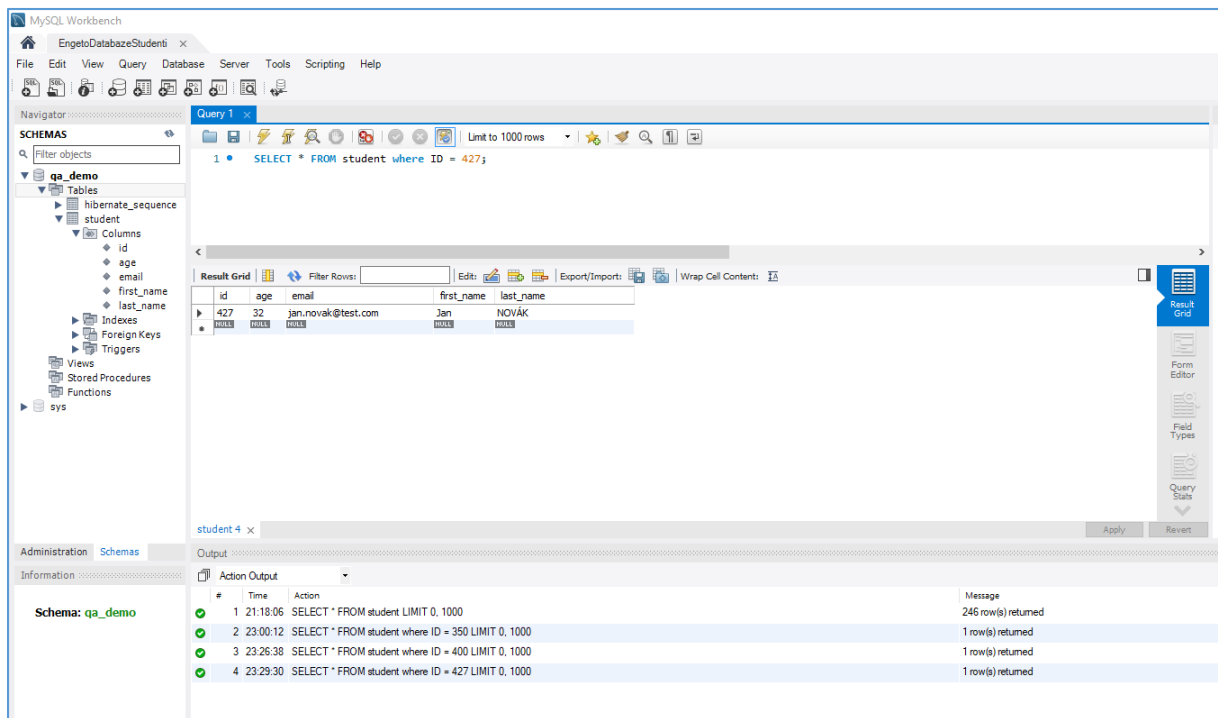
Zaměřeno na studenta s ID 427

1. Otevřela jsem si aplikaci Postman
 - a. „New“ → „http“ → roletka „GET“ → odkaz:
<http://108.143.193.45:8080/api/v1/students/427> → „Send“.



Obrázek 5: Postman: GET, ID studenta 427

2. Otevřela jsem aplikaci MySQLWorkbench
 - a. SQL příkaz: `SELECT * FROM student WHERE id = 427;`

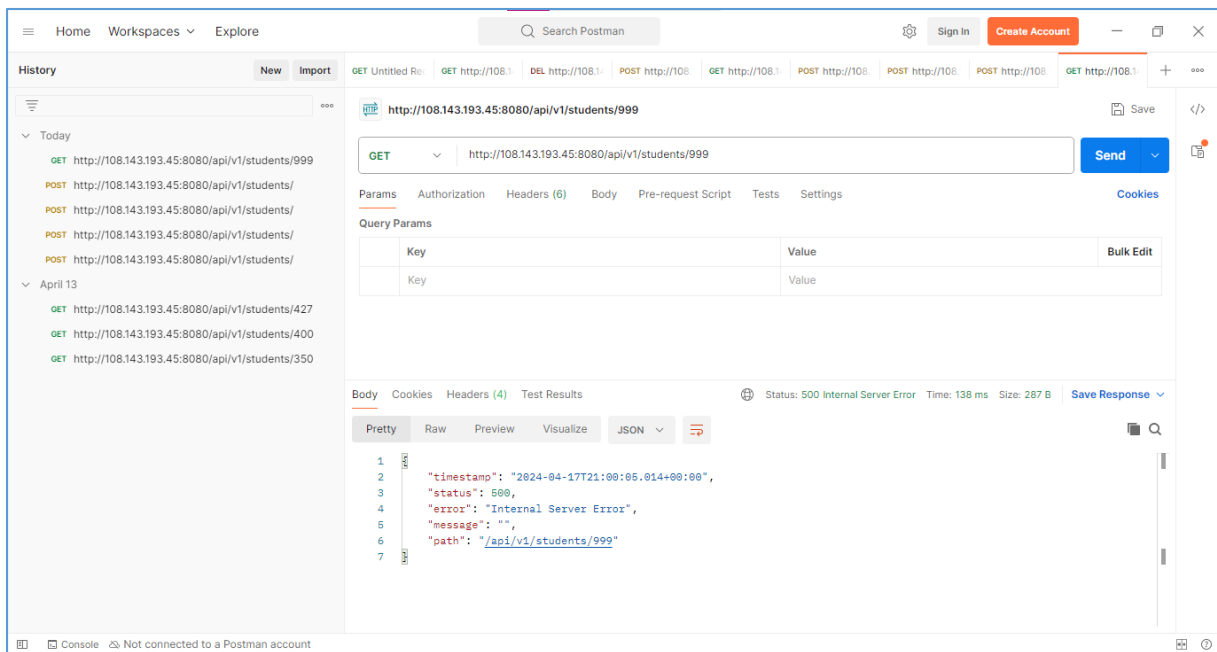


Obrázek 6: MSW: GET, ID studenta 427

- Očekávání:
 - o Status „200 OK“ – Potvrzeno.
 - o V databázi je vyhledán student s ID 427, kde jeho první jméno je Jan, příjmení NOVÁK, věk 32, email jan.novak@test.com
- Reálný výsledek: SOUHLASÍ, Chyba se neobjevila.

Zaměřeno na studenta s ID 999 – neexistující student

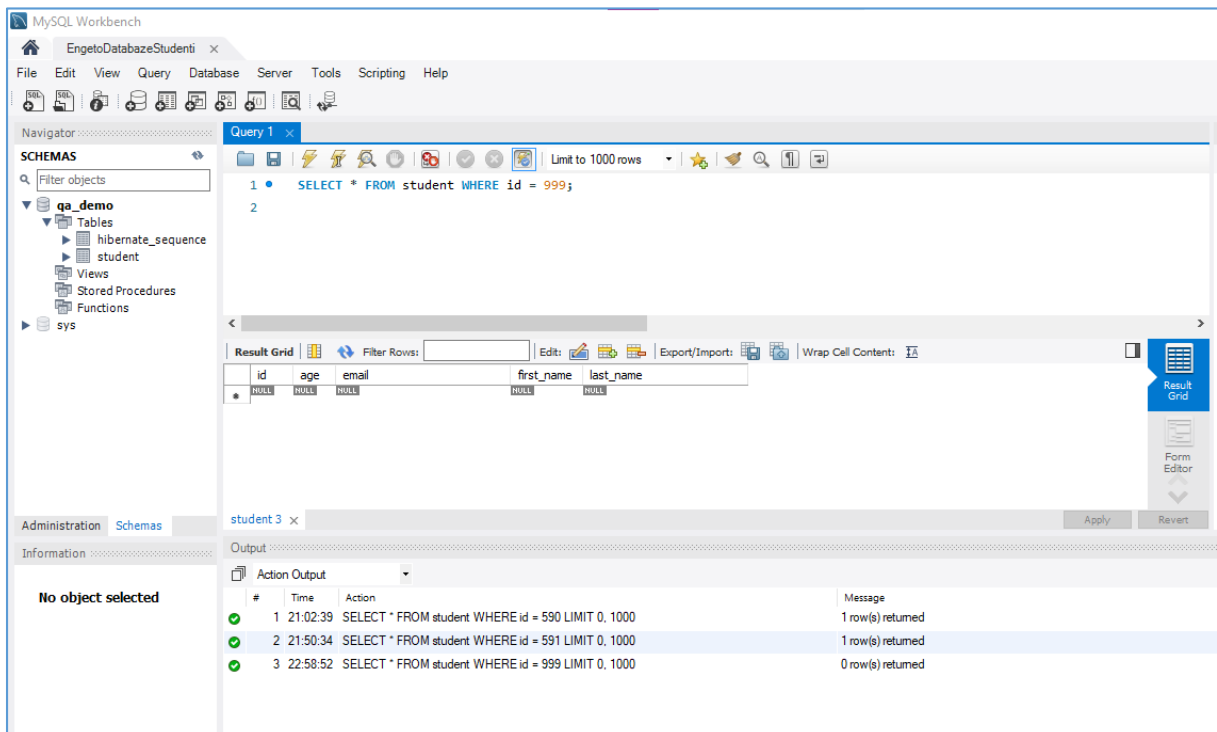
1. Otevřela jsem si aplikaci Postman
 - a. „New“ → „http“ → roletka „GET“ → odkaz:
<http://108.143.193.45:8080/api/v1/students/999> → „Send“.



Obrázek 7: Postman: GET, ID studenta 999 – neexistující student

2. Otevřela jsem aplikaci MySQLWorkbench

a. SQL příkaz: `SELECT * FROM student WHERE id = 999;`

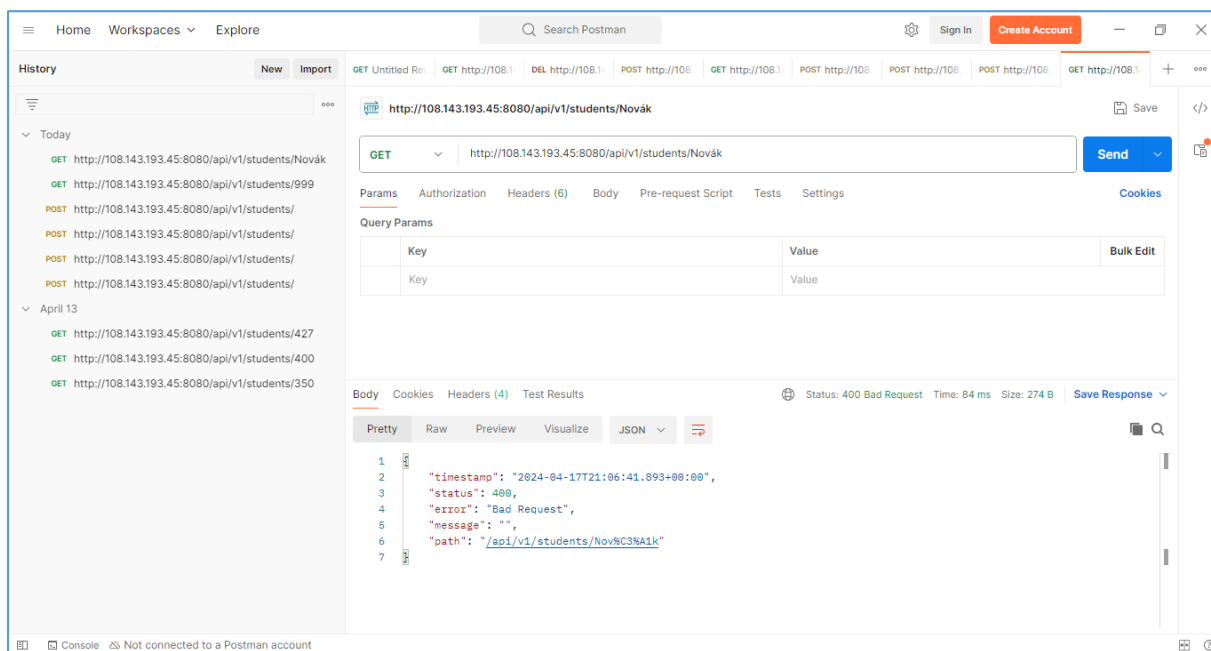


Obrázek 8: MSW: GET, ID studenta 999 - neexistující student

- Očekávání:
 - Status „200 OK“ – Nepotvrzeno.
 - Status „500 Internal Server Error“
- Reálný výsledek: V databázi není možné vyhledat studenta s ID 999.
 - Doplnit chybovou hlášku: „Lze vyhledávat jen v existujících záznamech“

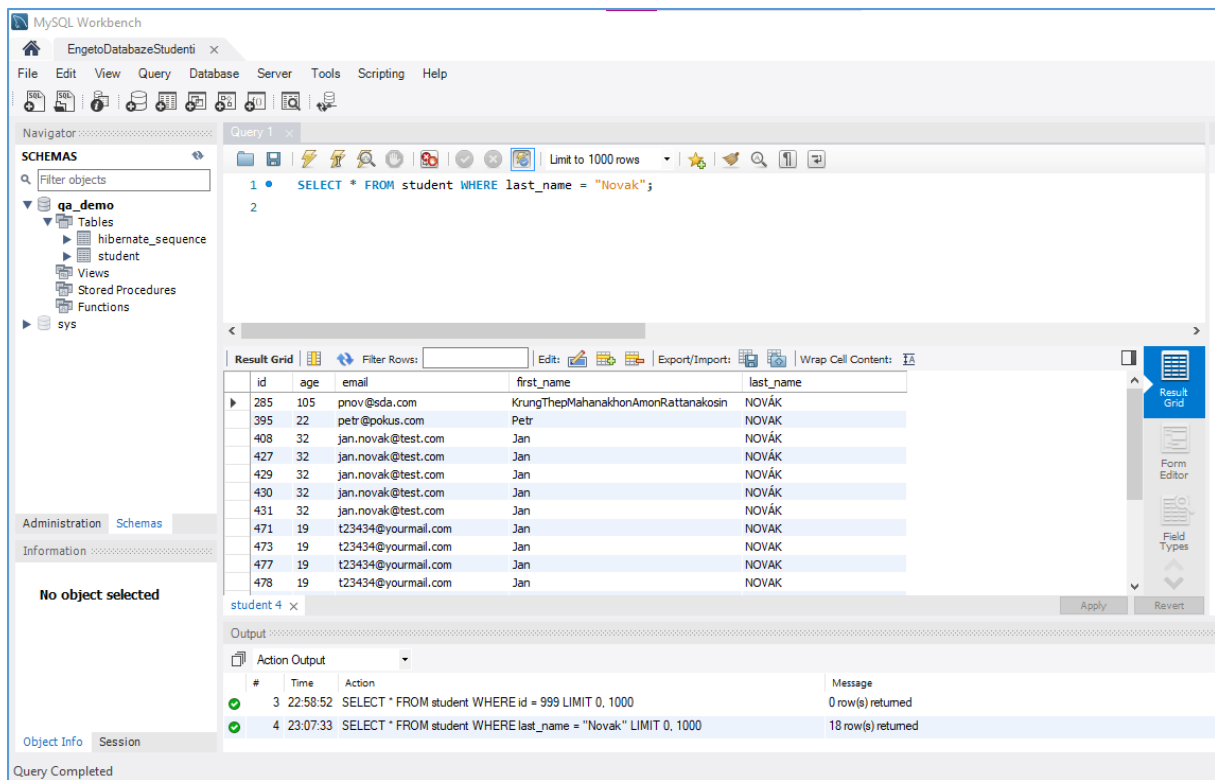
Zaměřeno na studenta „Novák“ – vícero stejných jmen

1. Otevřela jsem si aplikaci Postman
 - a. „New“ → „http“ → roletka „GET“ → odkaz:
<http://108.143.193.45:8080/api/v1/students/Novák> → „Send“.



Obrázek 9: Postman: GET, Novák - vícero stejných jmen

2. Otevřela jsem aplikaci MySQLWorkbench
 - a. SQL příkaz: `SELECT * FROM student WHERE last_name = „Novak“;`



Obrázek 10: MSW: GET, Novák - vícero stejných jmen

- Očekávání:
 - Status „200 OK“ – Nepotvrzeno.
 - Status „400 Bad Request“
- Reálný výsledek: V databázi se vyhledalo 18 řádků se jménem Novák
 - Doplnit chybovou hlášku: „Konkretizujte hledaná data“

POST

Dále jsem testovala funkci „Post“, která slouží k vytvoření nových dat. V tomto případě k vytvoření nového studenta. Pro testování jsem si náhodně vytvořila 5 nových studentů s různými obměnami jmen. Zkoušela jsem testovat standartní české jméno a příjmení ale poté jsem se zaměřila i na víceslovná jména, záporný věk, chyby v emailu, různé zkratky a tituly před i za jmény.

Vytvoření nového studenta Tereza Bocánová

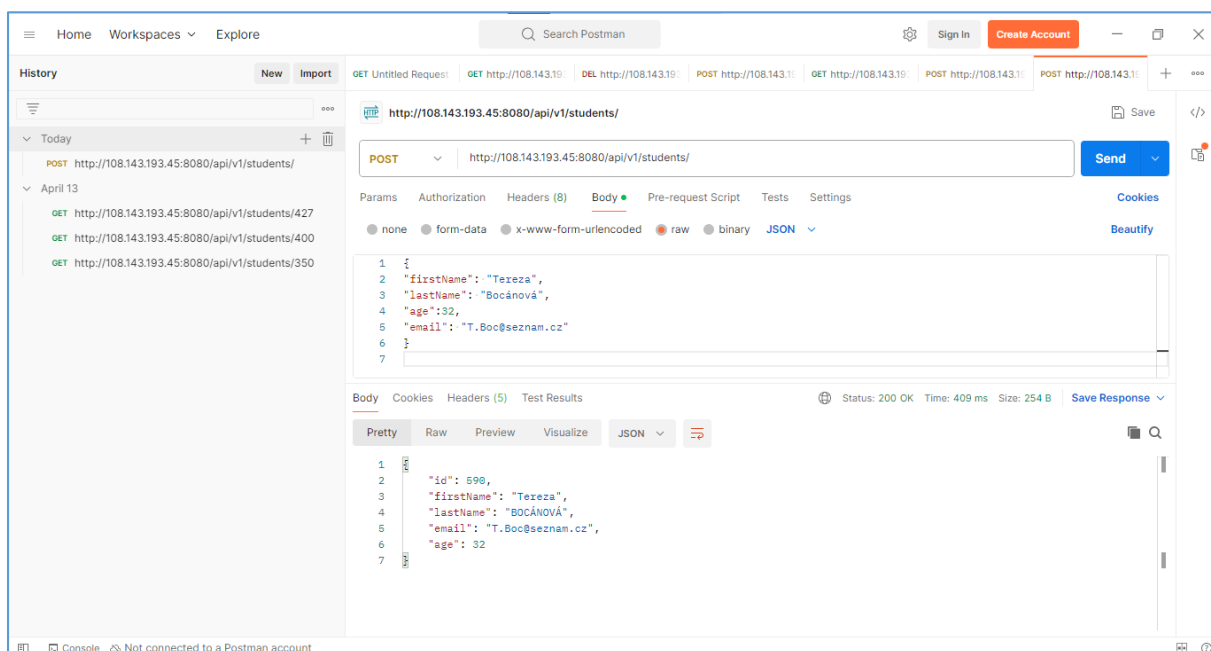
1. Otevřela jsem si aplikaci Postman

a. „New“ → „http“ → roletka „POST“ → odkaz:

<http://108.143.193.45:8080/api/v1/students/> → „Body“ → „raw“ → roletka „JSON“

```
1. {  
2.   "firstName": "Tereza",  
3.   "lastName": "Bocánová",  
4.   "age": 32,  
5.   "email": "T.Boc@seznam.cz"  
6. }
```

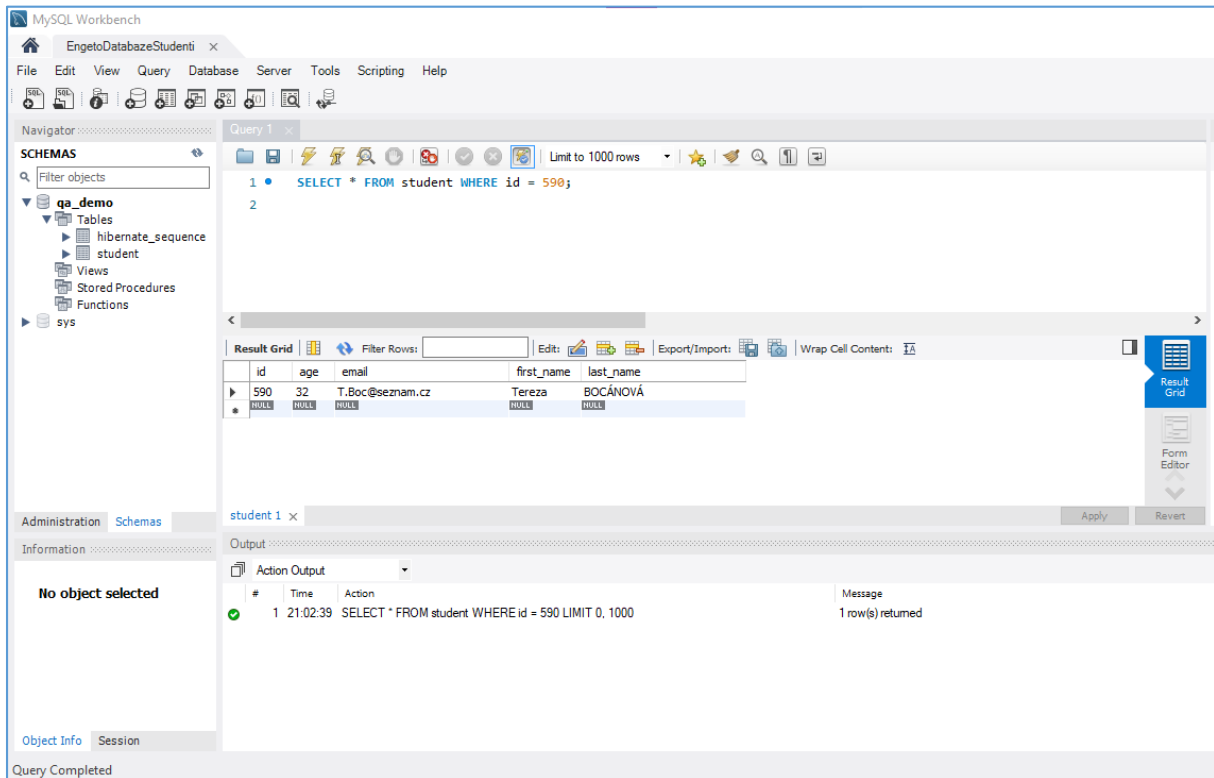
b. „Send“



Obrázek 11: Postman: POST, Tereza Bocánová

2. Otevřela jsem aplikaci MySQLWorkbench

a. SQL příkaz: `SELECT * FROM student WHERE id = 590;`



Obrázek 12: MSW: POST, ID studenta 590

- Očekávání:
 - o Status „200 OK“ – Potvrzeno.
 - o V databázi je vyhledán student s ID 590, kde jeho první jméno je Tereza, příjmení BOCÁNOVÁ, věk 32, email T.Boc@seznam.cz
- Nově zapsaný student je bez chyby
- Příjmení je dobře v databázi VELKÝMI PÍSMENY
- Reálný výsledek: SOUHLASÍ, Chyba se neobjevila.

Vytvořen nový student Anna-Maria Luisa Řehořčíčková

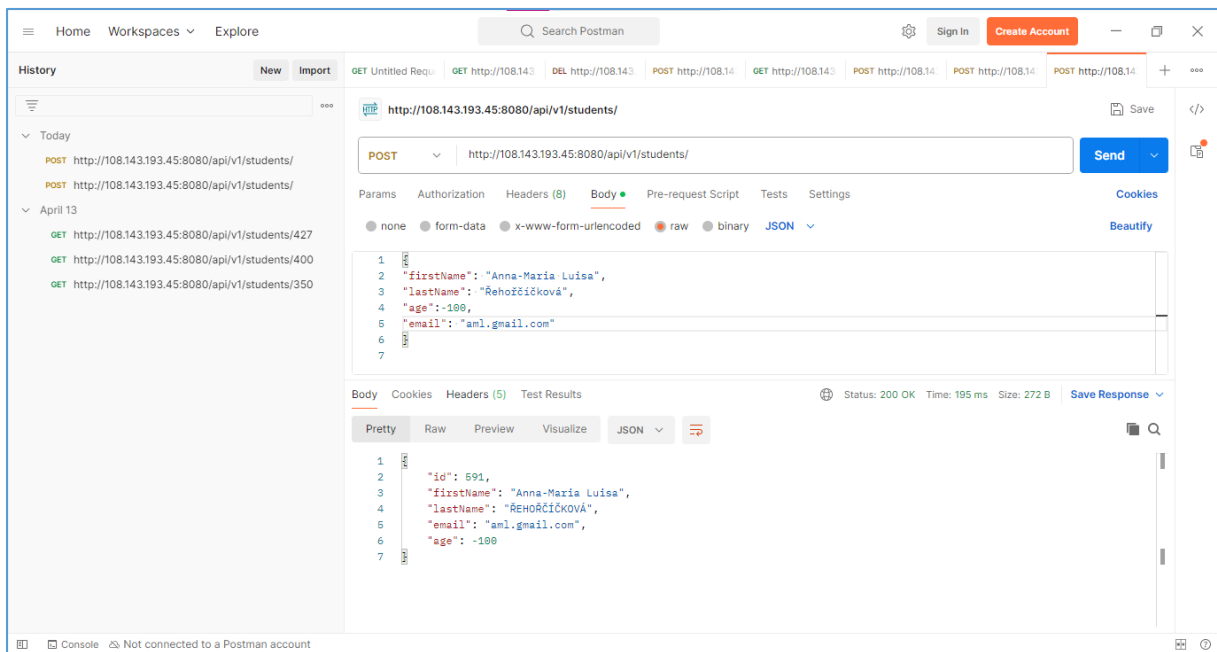
1. Otevřela jsem si aplikaci Postman

a. „New“ → „http“ → roletka „POST“ → odkaz:

<http://108.143.193.45:8080/api/v1/students/> → „Body“ → „raw“ → roletka „JSON“

```
1. {
2.   "firstName": "Anna-Maria Luisa",
3.   "lastName": "Řehořčíčková",
4.   "age": -100,
5.   "email": "aml@gmail.com"
6. }
```

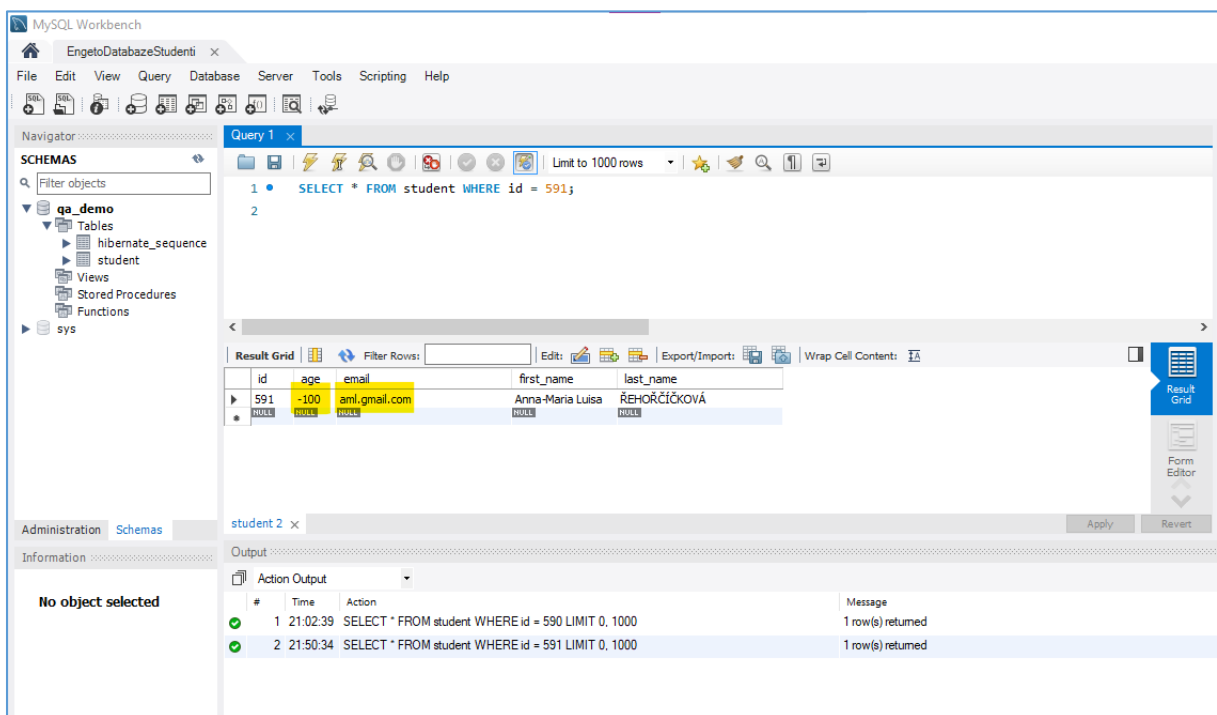
b. „Send“



Obrázek 13: Postman: POST, Anna-Maria Luisa Řehořčíčková

2. Otevřela jsem aplikaci MySQLWorkbench

a. SQL příkaz: `SELECT * FROM student WHERE id = 591;`



Obrázek 14: MSW: POST, Anna-Maria Luisa Řehořčíčková

- Očekávání:

- Status „200 OK“ – Potvrzeno.
- V databázi je vyhledán student s ID 591, kde jeho první jméno je Anna-Maria Luisa, příjmení Řehořčíčková, věk -100, email aml@gmail.com

- Nově zapsaný student je sice dle požadavků přes Postmana, ale očekávala bych upozornění na chybně zadaný věk a email.
 - o Věk: Chybová hláška: „Zadejte kladné číslo!“
 - o Email: „Chybně zadaný formát emailu!“
- Příjmení je dobře v databázi VELKÝMI PÍSMENY
- Reálný výsledek sice souhlasí, ale nedává smysl = BUG!

Vytvořen nový student jana ze 3.patra 123+-*/*@

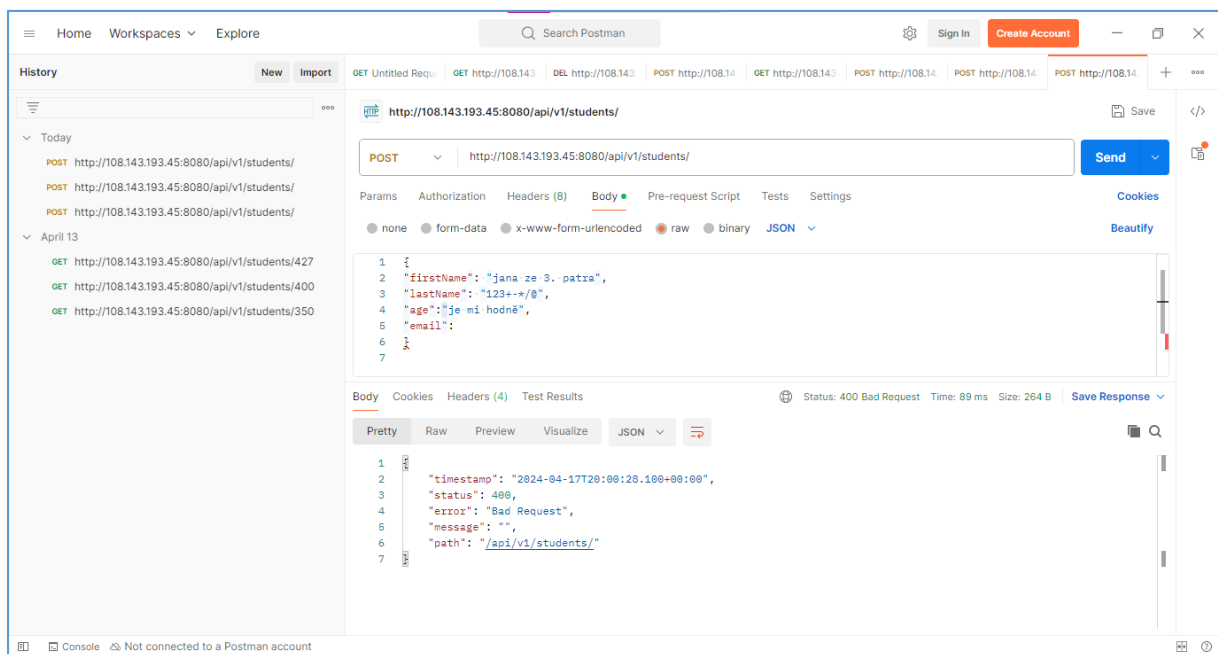
1. Otevřela jsem si aplikaci Postman

a. „New“ → „http“ → roletka „POST“ → odkaz:

<http://108.143.193.45:8080/api/v1/students/> → „Body“ → „raw“ → roletka „JSON“

```
1. {
2.   "firstName": "jana ze 3. patra",
3.   "lastName": "123+-*/*@"
4.   "age": "je mi hodně",
5.   "email":
6. }
```

b. „Send“



Obrázek 15: Postman: POST, jana ze 3.patra 123+-*/*@

- Očekávání:

- o Status: 200 OK – nepotvrzeno.
 - Status „400 Bad Request“
- o Nově zapsaný student: „jana ze 3.patra 123+-*/*@“, věk: „je mi hodně“, email: „“ .
 - Skutečnost: Chybová hláška – nepodařilo se zadat do databáze.

- Doplnit chybové hlášky již při zadávání studenta do databáze.
 - First name: „Zadejte jméno!“
 - Last name: „Zadejte příjmení!“
 - Věk: „Zadejte kladné číslo!“
 - Email: „Chybně zadaný formát emailu!“

Vytvořen nový student Mgr. Ing. Jan III. Müller L' de Foňt PhD.

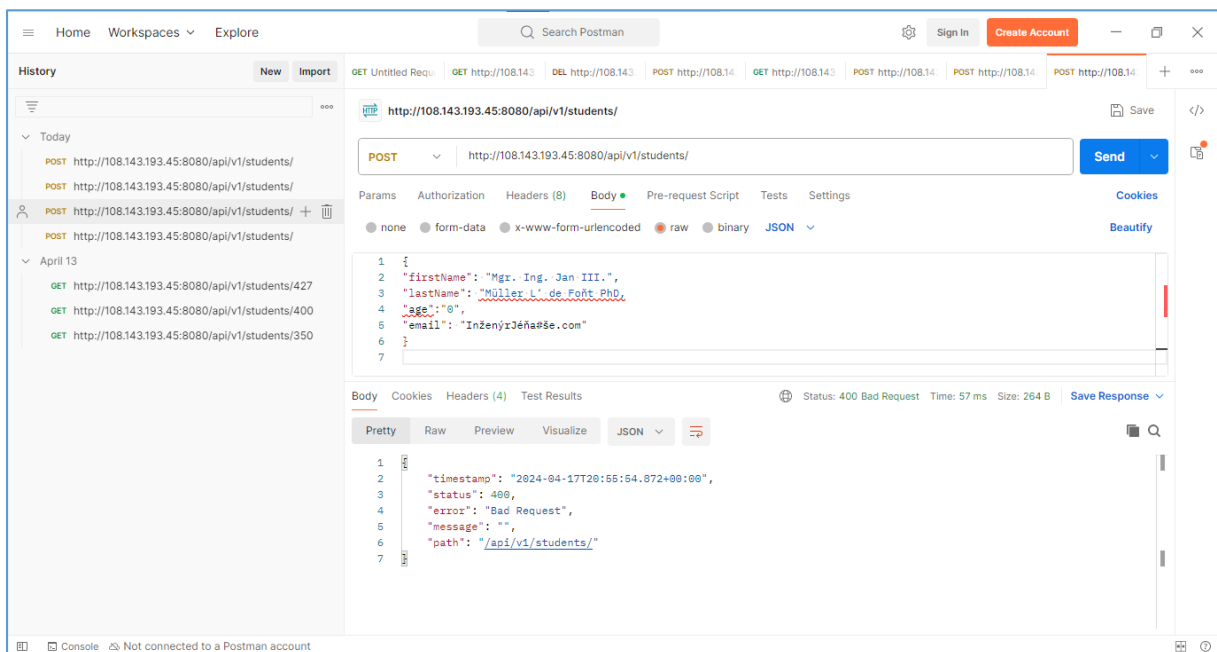
1. Otevřela jsem si aplikaci Postman

a. „New“ → „http“ → roletka „POST“ → odkaz:

<http://108.143.193.45:8080/api/v1/students/> → „Body“ → „raw“ → roletka „JSON“

```
1. {
2.   "firstName": "Mgr. Ing. Jan III.",
3.   "lastName": "Müller L' de Foňt PhD"
4.   "age": "0",
5.   "email": "InženýrJěňa#še.com"
6. }
```

b. „Send“



Obrázek 16: Postman: POST, Mgr. Ing. Jan III. Müller L' de Foňt PhD.

- Očekávání:

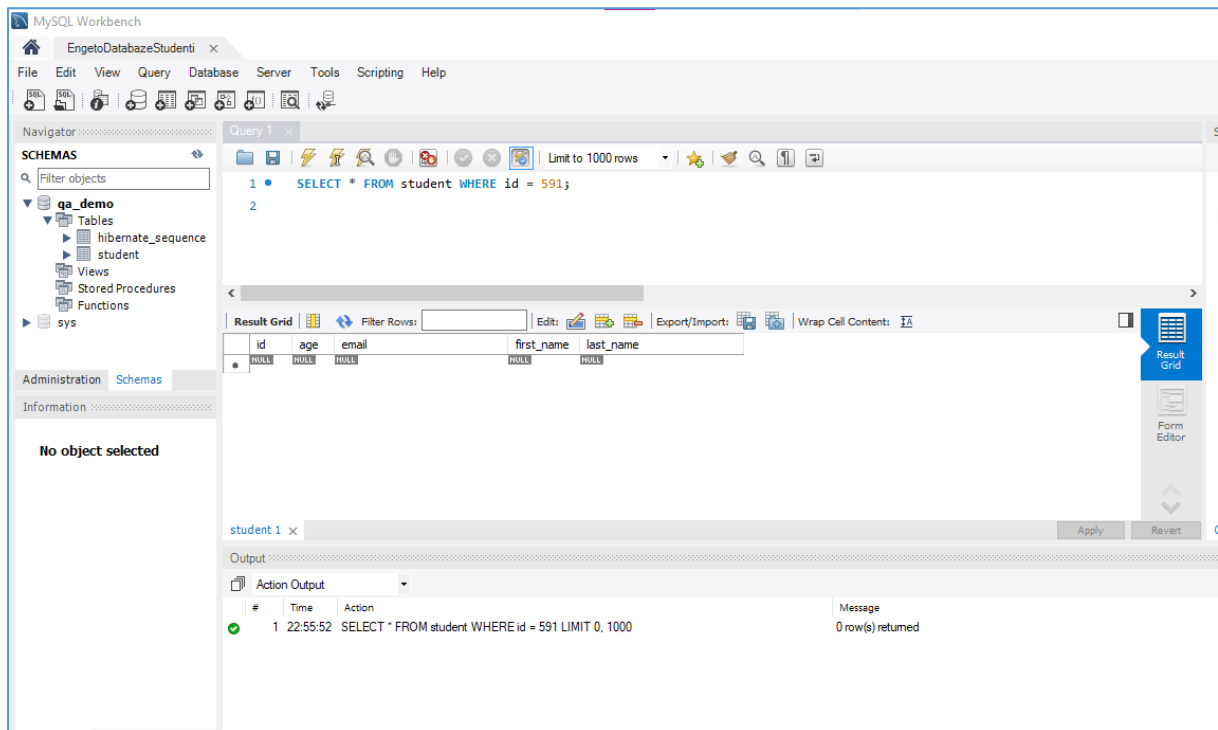
- Status: 200 OK – nepotvrzeno.
 - Status „400 Bad Request“
- Nově zapsaný student: „Mgr. Ing. Jan III. Müller L' de Foňt PhD., věk: „0“, email: „InženýrJěňa#še.com“.
- Skutečnost: Chybová hláška – nepodařilo se zadat do databáze.
- Doplnit chybové hlášky již při zadávání studenta do databáze.
 - First name: „Zadejte pouze jméno!“
 - Last name: „Zadejte pouze příjmení!“
 - Věk: „Zadejte kladné číslo!“
 - Email: „Chybně zadaný formát emailu!“

DELETE

Dále jsem testovala funkci „Delete“, která slouží ke smazání dat v databázi.

Smazán student s ID 591

1. Otevřela jsem si aplikaci Postman
 - a. „New“ → „http“ → roletka „DELETE“ → odkaz:
<http://108.143.193.45:8080/api/v1/students/591> → „Send“
2. Otevřela jsem aplikaci MySQLWorkbench
 - b. SQL příkaz: `SELECT * FROM student WHERE id = 591;`

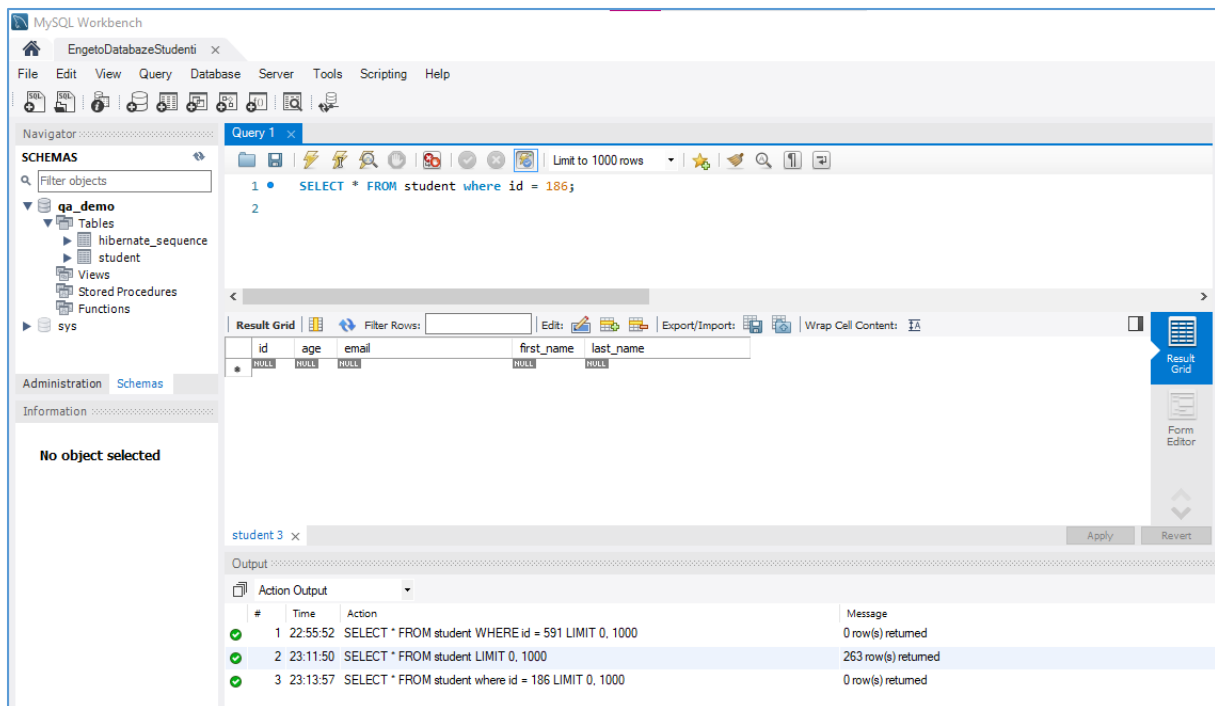


Obrázek 17: MSW: DELETE student s ID 591

- Očekávání:
 - o Status: 200 OK – potvrzeno.
 - o Student s ID 591 úspěšně smazán.
- Reálný výsledek: SOUHLASÍ, Chyba se neobjevila.

Smazán student s ID 186

1. Otevřela jsem si aplikaci Postman
 - a. „New“ → „http“ → roletka „DELETE“ → odkaz:
<http://108.143.193.45:8080/api/v1/students/186> → „Send“
2. Otevřela jsem aplikaci MySQLWorkbench
 - c. SQL příkaz: `SELECT * FROM student WHERE id = 186;`

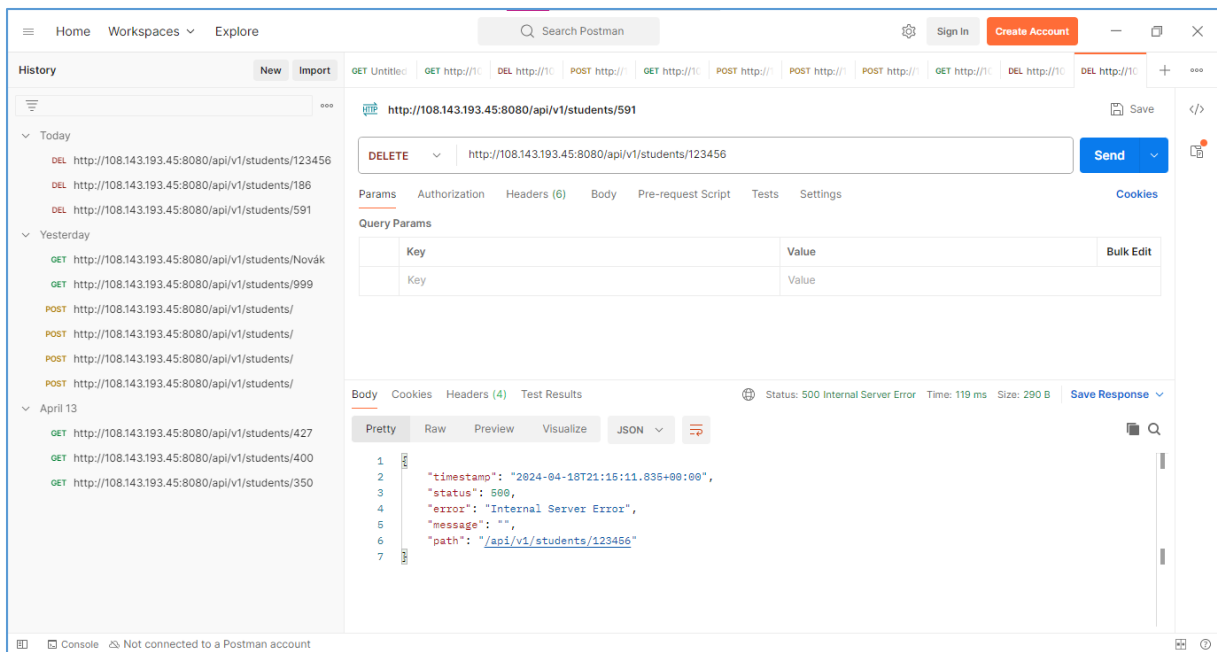


Obrázek 18: MSW: DELETE studen s ID 186

- Očekávání:
 - o Status: 200 OK – potvrzeno.
 - o Student s ID 186 úspěšně smazán.
- Reálný výsledek: SOUHLASÍ, Chyba se neobjevila.

Smazán student s ID 123456 – neexistující student

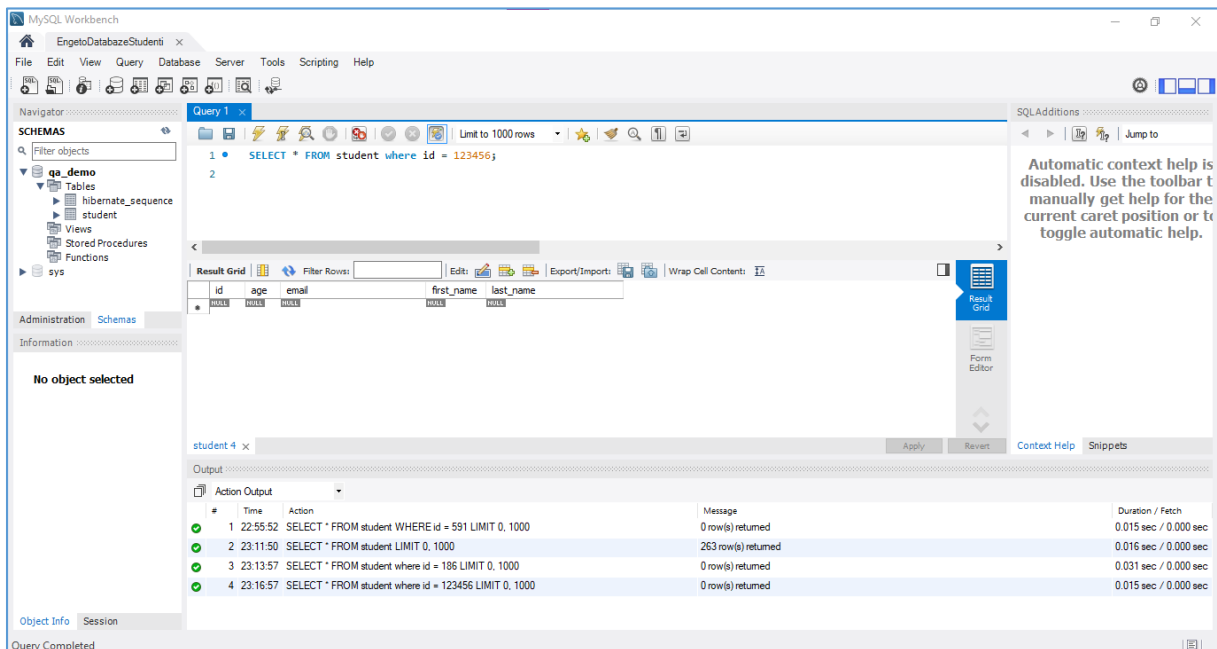
1. Otevřela jsem si aplikaci Postman
 - a. „New“ → „http“ → roletka „DELETE“ → odkaz: <http://108.143.193.45:8080/api/v1/students/123456> → „Send“



Obrázek 19: Postman: DELETE, student s ID 123456 - neexistující student

2. Otevřela jsem aplikaci MySQLWorkbench

d. SQL příkaz: `SELECT * FROM student WHERE id = 123456;`



Obrázek 20: MSW: DELETE, student s ID 123456 - neexistující student

- Očekávání:
 - Status: 200 OK – nepotvrzeno!
 - Status: 500 Internal Server Error
 - Student s ID 123456 neexistuje v databázi – nelze smazat.
- Reálný výsledek: Nelze smazat z databáze studenta, který tam ani není
 - Doplnit chybovou hlášku
 - „Nelze smazat neexistující záznam!“

ZÁVĚR

Na tomto projektu jsem si vyzkoušela vyhledat v databázi studenty, následně jak jejich tvorbu, tak i vymazání z databáze. Z jednoduchých testovacích scénářů nebyly vyvozeny žádné chyby, nicméně ve složitějších scénářích se chyby již objevily. U nalezených chyb jsem doplnila očekávání na případnou opravu.