

FINÁLNÍ PROJEKT

č.1



Autor: Pavel Formánek
Datum: 9.6. 2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	11
BUG REPORT	28

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: *****
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.

Požadavky před testováním:

- 1) Mít nainstalovaný OS Windows 10
- 2) Mít přístup k databázi SQL s tabulkou student, použití programu MySQL Workbench
- 3) Nainstalovaný program Postman
- 4) Logická posloupnost – nejdříve testování metody POST, poté testování metody GET, nakonec testování metody DELETE

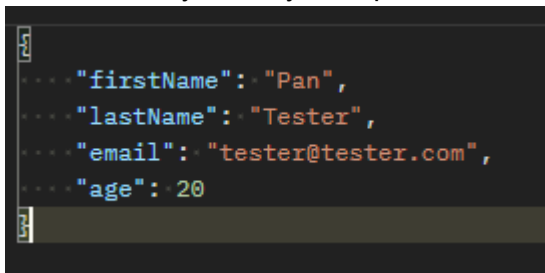
1) METODA POST

1. TESTOVACÍ SCÉNÁŘ

Abstrakt – Ověření metodou POST založení nového studenta v databázi přes REST – API

KROKY:

- 1) Spustit aplikaci Postman
- 2) Vybrat v nabídce tlačítko New , následně z nabídky HTTP a zvolit metodu POST
- 3) Vybrat v nabídce BODY , zaškrtnout políčko raw a vybrat formát JSON
- 4) Zadat všechny atributy a doplnit do nich tyto hodnoty:



```
{
  "firstName": "Pan",
  "lastName": "Tester",
  "email": "tester@tester.com",
  "age": 20
}
```

- 5) Kliknout na tlačítko SEND a tím odeslat POST požadavek na adresu <http://108.143.193.45:8080/api/v1/students/>
- 6) Zkontrolovat status code v programu Postman – 201 Created
- 7) Zkontrolovat výstup aplikace Postman – BODY, Pretty a formát JSON
- 8) Zkontrolovat, zda bylo studentovi přiděleno náhodné ID číslo, zda se v údajích shoduje odpověď z databáze s daty z požadavku POST.

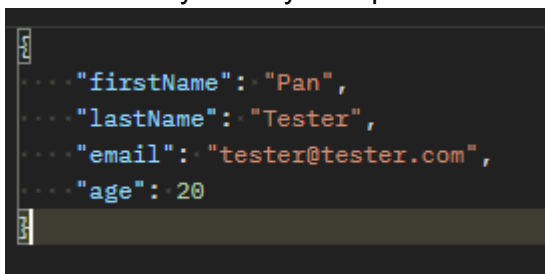
- 9) Zkontrolovat a zaznamenat, zda se nachází student v databázi spolu s uvedenými atributy a hodnotami s použitím nástroje MySQL WorkBench

2. TESTOVACÍ SCÉNÁŘ

Abstrakt – Ověření metodou POST, zda je možné duplicitní založení stejného studenta jako bylo v 1. Testovacím scénáři, do databáze přes REST – API

KROKY:

- 1) Spustit aplikaci Postman
- 2) Vybrat v nabídce tlačítko New , následně z nabídky HTTP a zvolit metodu POST
- 3) Vybrat v nabídce BODY , zaškrtnout políčko raw a vybrat formát JSON
- 4) Zadat všechny atributy a doplnit do nich tyto hodnoty:



```
{
  "firstName": "Pan",
  "lastName": "Tester",
  "email": "tester@tester.com",
  "age": 20
}
```

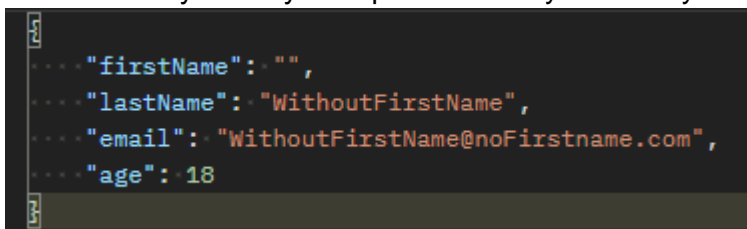
- 5) Kliknout na tlačítko SEND a tím odeslat POST požadavek na adresu <http://108.143.193.45:8080/api/v1/students/>
- 6) Zkontrolovat status code v programu Postman – 201 Created
- 7) Zkontrolovat výstup aplikace Postman – BODY, Pretty a formát JSON
- 8) Zkontrolovat, zda bylo studentovi přiděleno náhodné ID číslo, zda se v údajích shoduje odpověď z databáze s daty z požadavku POST.
- 9) Zkontrolovat a zaznamenat, zda se nachází student v databázi spolu s uvedenými atributy a hodnotami s použitím nástroje MySQL WorkBench

3. TESTOVACÍ SCÉNÁŘ

Abstrakt – Ověření metodou POST založení nového studenta bez zadání hodnoty do atributu “firstName” v databázi přes REST – API

KROKY:

- 1) Spustit aplikaci Postman
- 2) Vybrat v nabídce tlačítko New , následně z nabídky HTTP a zvolit metodu POST
- 3) Vybrat v nabídce BODY , zaškrtnout políčko raw a vybrat formát JSON
- 4) Zadat všechny atributy a doplnit do nich tyto hodnoty:



```
...."firstName": "",  
...."lastName": "WithoutFirstName",  
...."email": "WithoutFirstName@noFirstname.com",  
...."age": 18
```

- 5) Kliknout na tlačítko SEND a tím odeslat POST požadavek na adresu <http://108.143.193.45:8080/api/v1/students/>
- 6) Zkontrolovat status code v programu Postman – 201 Created
- 7) Zkontrolovat výstup aplikace Postman – BODY, Pretty a formát JSON
- 8) Zkontrolovat, zda bylo studentovi přiděleno náhodné ID číslo, zda se v údajích shoduje odpověď z databáze s daty z požadavku POST.
- 9) Zkontrolovat a zaznamenat, zda se nachází student v databázi spolu s uvedenými atributy a hodnotami s použitím nástroje MySQL WorkBench

4. TESTOVACÍ SCÉNÁŘ

Abstrakt – Ověření metodou POST založení nového studenta v databázi přes REST – API, se zadáním hodnoty do atributu “firstName” s použitím diakritiky

KROKY:

- 1) Spustit aplikaci Postman
- 2) Vybrat v nabídce tlačítko New , následně z nabídky HTTP a zvolit metodu POST
- 3) Vybrat v nabídce BODY , zaškrtnout políčko raw a vybrat formát JSON
- 4) Zadat všechny atributy a doplnit do nich tyto hodnoty:

```
{
  "firstName": "Štěpánek",
  "lastName": "Maličkatý",
  "email": "maličkatýštěpánek@seznamek.čsfr",
  "age": 1
}
```

- 5) Kliknout na tlačítko SEND a tím odeslat POST požadavek na adresu <http://108.143.193.45:8080/api/v1/students/>
- 6) Zkontrolovat status code v programu Postman – 201 Created
- 7) Zkontrolovat výstup aplikace Postman – BODY, Pretty a formát JSON
- 8) Zkontrolovat, zda bylo studentovi přiděleno náhodné ID číslo, zda se v údajích shoduje odpověď z databáze s daty z požadavku POST.
- 9) Zkontrolovat a zaznamenat, zda se nachází student v databázi spolu s uvedenými atributy a hodnotami s použitím nástroje MySQL WorkBench

2) METODA GET

5. TESTOVACÍ SCÉNÁŘ

Abstrakt – Získání dat z databáze metodou GET přes REST – API, zadané ID studenta je 490

KROKY:

- 1) Spustit aplikaci Postman
- 2) Vybrat v nabídce tlačítko New , následně z nabídky HTTP a zvolit metodu GET
- 3) Zadat adresu <http://108.143.193.45:8080/api/v1/students/>
- 4) Za adresu přidat ID studenta 490
- 5) Požadavek bude vyplněn takto <http://108.143.193.45:8080/api/v1/students/490>
- 6) Kliknout na tlačítko GET a tím odeslat GET požadavek na adresu uvedenou výše
- 7) Zkontrolovat status code v programu Postman – 200 OK
- 8) Zkontrolovat výstup aplikace Postman – BODY
- 9) Zkontrolovat a zaznamenat, zda se shodují všechna data o studentovi zobrazená v aplikaci Postman s veškerými daty v databázi spolu s uvedenými atributy a hodnotami s použitím nástroje MySQL WorkBench

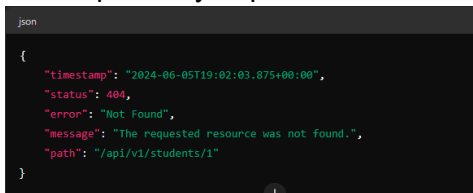
6. TESTOVACÍ SCÉNÁŘ

Abstrakt – Získání dat z databáze metodou GET přes REST – API, zadané ID studenta je 1 (ID 1 – neexistující záznam s tímto ID v databázi)

KROKY:

- 1) Spustit aplikaci Postman
- 2) Vybrat v nabídce tlačítko New , následně z nabídky HTTP a zvolit metodu GET
- 3) Zadat adresu <http://108.143.193.45:8080/api/v1/students/>
- 4) Za adresu přidat ID studenta 1 (ID 1 se v databázi nenachází)
- 5) Požadavek bude vyplněn takto <http://108.143.193.45:8080/api/v1/students/1>

- 6) Kliknout na tlačítko GET a tím odeslat GET požadavek na adresu uvedenou výše
- 7) Zkontrolovat status code v programu Postman – 404 Not Found
- 8) Zkontrolovat výstup aplikace Postman – BODY
- 9) Zkontrolovat a zaznamenat, že se student s tímto ID 1 opravdu nenachází v databázi s použitím nástroje MySQL WorkBench a výstupní data v aplikaci Postman mají formát podobný např. tomuto :



```
json
{
  "timestamp": "2024-06-05T19:02:03.875+00:00",
  "status": 404,
  "error": "Not Found",
  "message": "The requested resource was not found.",
  "path": "/api/v1/students/1"
}
```

7. TESTOVACÍ SCÉNÁŘ

Abstrakt – Získání dat z databáze metodou GET přes REST – API, zadané ID studenta je 1152

KROKY:

- 1) Spustit aplikaci Postman
- 2) Vybrat v nabídce tlačítko New , následně z nabídky HTTP a zvolit metodu GET
- 3) Zadat adresu <http://108.143.193.45:8080/api/v1/students/>
- 4) Za adresu přidat ID studenta 1152
- 5) Požadavek bude vyplněn takto <http://108.143.193.45:8080/api/v1/students/1152>
- 6) Kliknout na tlačítko GET a tím odeslat GET požadavek na adresu uvedenou výše
- 7) Zkontrolovat status code v programu Postman – 200 OK
- 8) Zkontrolovat výstup aplikace Postman – BODY
- 9) Zkontrolovat a zaznamenat, že se student s tímto ID 1152 se nachází v databázi s použitím nástroje MySQL WorkBench a výstupní data se shodují s odpovědí aplikaci Postman

3) METODA DELETE

8. TESTOVACÍ SCÉNÁŘ

Abstrakt – Smazání studenta v databázi přes REST – API, student se zadaným ID 0 (“nula”) se nenachází v databázi

KROKY:

- 1) Spustit aplikaci Postman
- 2) Vybrat v nabídce tlačítko New , následně z nabídky HTTP a zvolit metodu GET
- 3) Zadat adresu <http://108.143.193.45:8080/api/v1/students/>
- 4) Za adresu přidat ID studenta 0 (“nula”)
- 5) Požadavek bude vyplněn takto <http://108.143.193.45:8080/api/v1/students/0>
- 6) Kliknout na tlačítko DELETE a tím odeslat DELETE požadavek na adresu uvedenou výše
- 7) Zkontrolovat status code v programu Postman – 404 Not Found
- 8) Zkontrolovat výstup aplikace Postman – BODY
- 9) Zkontrolovat a zaznamenat, že se student s tímto ID 0 (“nula”) se nenachází v databázi s použitím nástroje MySQL WorkBench a výstupní data se shodují s odpovědí aplikace Postman

9. TESTOVACÍ SCÉNÁŘ

Abstrakt – Smazání studenta v databázi přes REST – API, student se zadaným ID 1120

KROKY:

- 1) Spustit aplikaci Postman
- 2) Vybrat v nabídce tlačítko New , následně z nabídky HTTP a zvolit metodu GET
- 3) Zadat adresu <http://108.143.193.45:8080/api/v1/students/>
- 4) Za adresu přidat ID studenta 1120
- 5) Požadavek bude vyplněn takto <http://108.143.193.45:8080/api/v1/students/1120>
- 6) Kliknout na tlačítko DELETE a tím odeslat DELETE požadavek na adresu uvedenou výše
- 7) Zkontrolovat status code v programu Postman – 200 OK nebo 204 No Content
- 8) Zkontrolovat výstup aplikace Postman – BODY
- 9) Zkontrolovat a zaznamenat, že se student s tímto ID 1120 se nenachází v databázi s použitím nástroje MySQL WorkBench a výstupní data se shodují s odpovědí aplikace Postman

10. TESTOVACÍ SCÉNÁŘ

Abstrakt – Smazání studenta v databázi přes REST – API, student se zadaným ID 1156

KROKY:

- 1) Spustit aplikaci Postman
- 2) Vybrat v nabídce tlačítko New , následně z nabídky HTTP a zvolit metodu GET
- 3) Zadat adresu <http://108.143.193.45:8080/api/v1/students/>
- 4) Za adresu přidat ID studenta 1156
- 5) Požadavek bude vyplněn takto <http://108.143.193.45:8080/api/v1/students/1156>
- 6) Kliknout na tlačítko DELETE a tím odeslat DELETE požadavek na adresu uvedenou výše
- 7) Zkontrolovat status code v programu Postman – 200 OK nebo 204 No Content
- 8) Zkontrolovat výstup aplikace Postman – BODY
- 9) Zkontrolovat a zaznamenat, že se student s tímto ID 1120 se nenachází v databázi s použitím nástroje MySQL WorkBench a výstupní data se shodují s odpovědí aplikace Postman

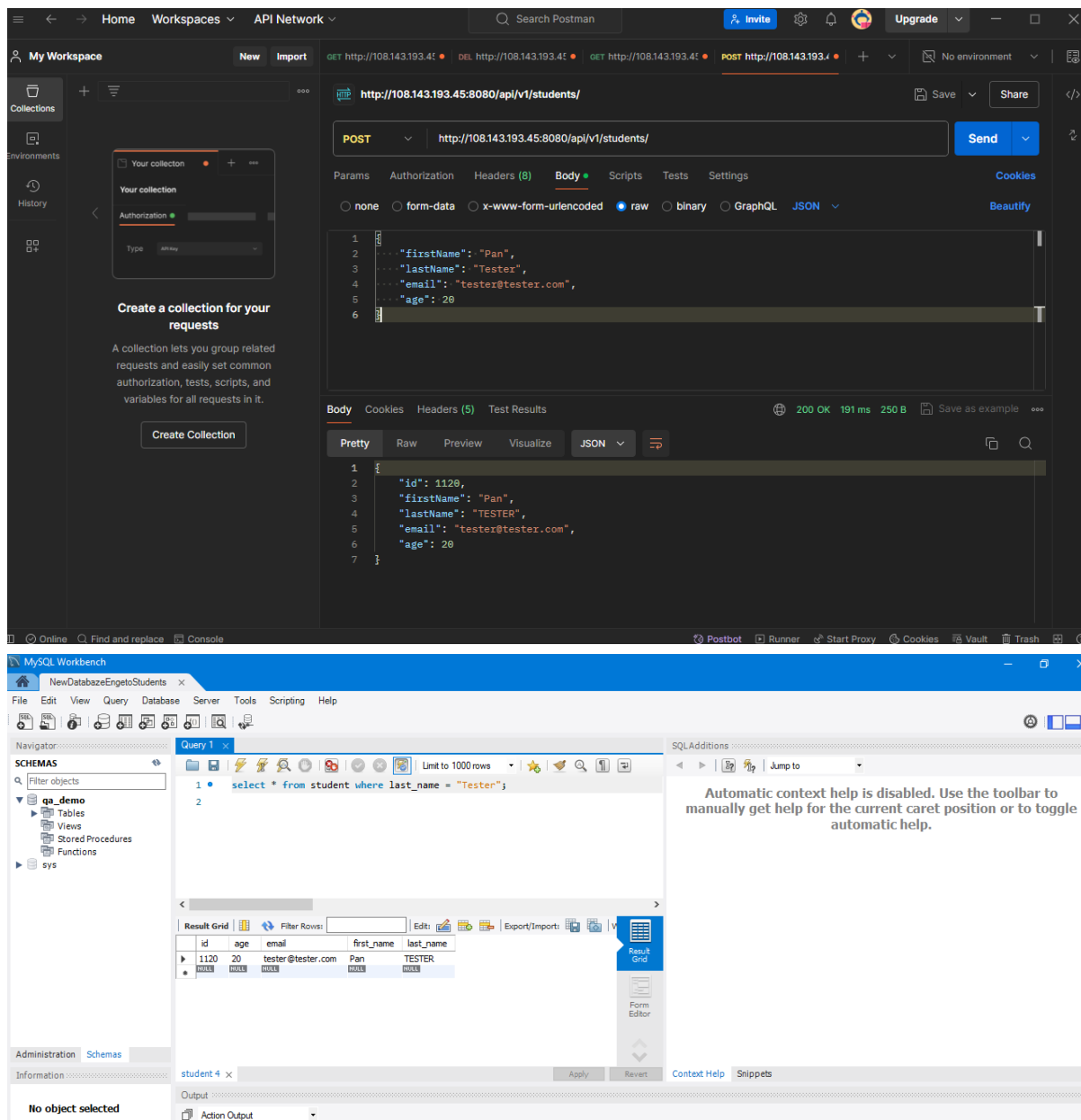
EXEKUCE TESTŮ

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

1) METODA POST

1. Exekuce testovacího scénáře - výsledky

Abstrakt – Ověření metodou POST založení nového studenta s náhodně vygenerovaným ID číslem v databázi přes REST – API



Skutečné výsledky: Při provedení metody POST se založilo studentovi ID s číslem 1120.

Atributy `firstName`, `lastName`, `email` a `age` byly shodné jak s vyslaným požadavkem, tak se záznamem v databázi.

Hodnoty `firstName`, `email` a `age` byli shodné jak s vyslaným požadavkem, tak se záznamem v databázi.

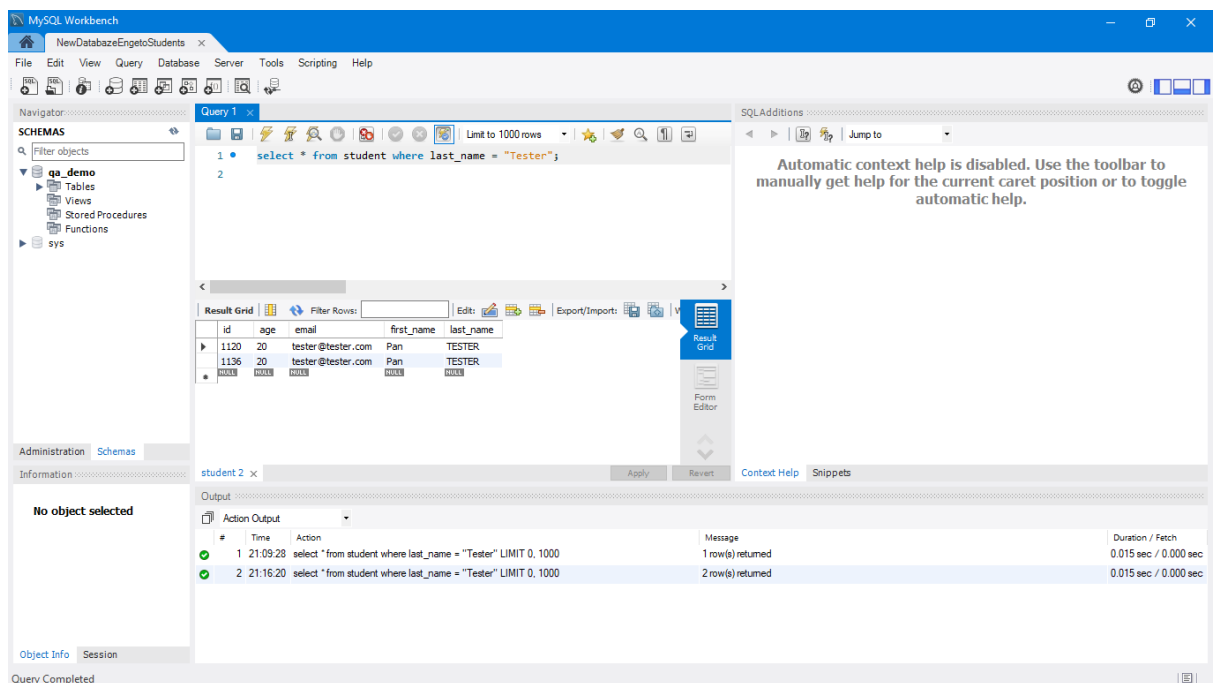
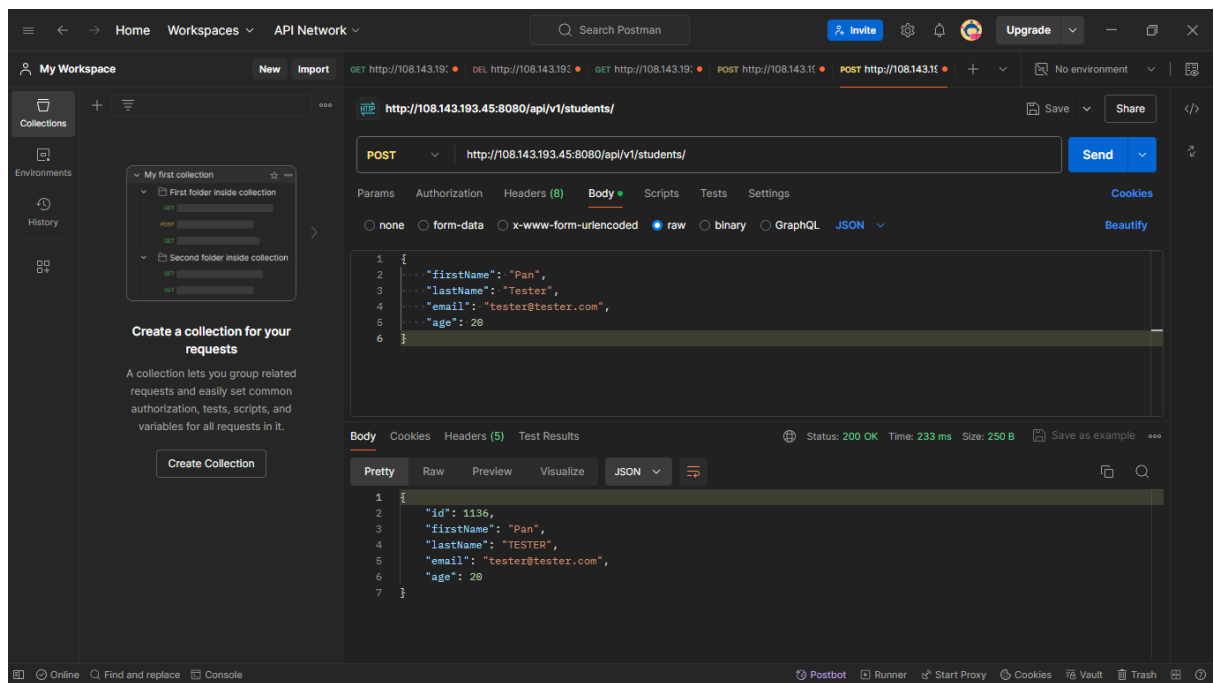
Hodnota lastName se lišila s vyslaným požadavkem, kdy byl vyslán požadavek lastName: "Tester" a výsledek byl lastName: "TESTER" .
Této chybě bylo přiděleno ID 1.

Status Code byl 200, očekávaný byl 201. Této chybě bylo přiděleno ID 2.

Očekávané výsledky: Očekávala se shoda jak v attributech, tak i v hodnotách – vyslaných v požadavcích = uloženými v databázi.
Očekávalo se náhodně vygenerované číslo ID studenta.
Očekával se status code 201.

2. Exekuce testovacího scénáře - výsledky

Abstrakt – Ověření metodou POST, zda je možné duplicitní založení stejného studenta jako bylo v 1. Testovacím scénáři, do databáze přes REST – API



Skutečné výsledky: Při provedení metody POST, při které byly použity úplně stejné hodnoty jako v 1. testovacím případě, se založilo studentovi ID s číslem 1136. Byla vytvořena duplicita. Atributy firstName, lastName, email a age byly shodné jak s vyslaným požadavkem, tak se záznamem v databázi.

Hodnoty firstName, email a age byli shodné jak s vyslaným požadavkem, tak se záznamem v databázi.

Hodnota lastName se lišila s vyslaným požadavkem, kdy byl vyslán požadavek lastName: "Tester" a výsledek byl lastName: "TESTER " .
Této chybě bylo již přiděleno ID 1.

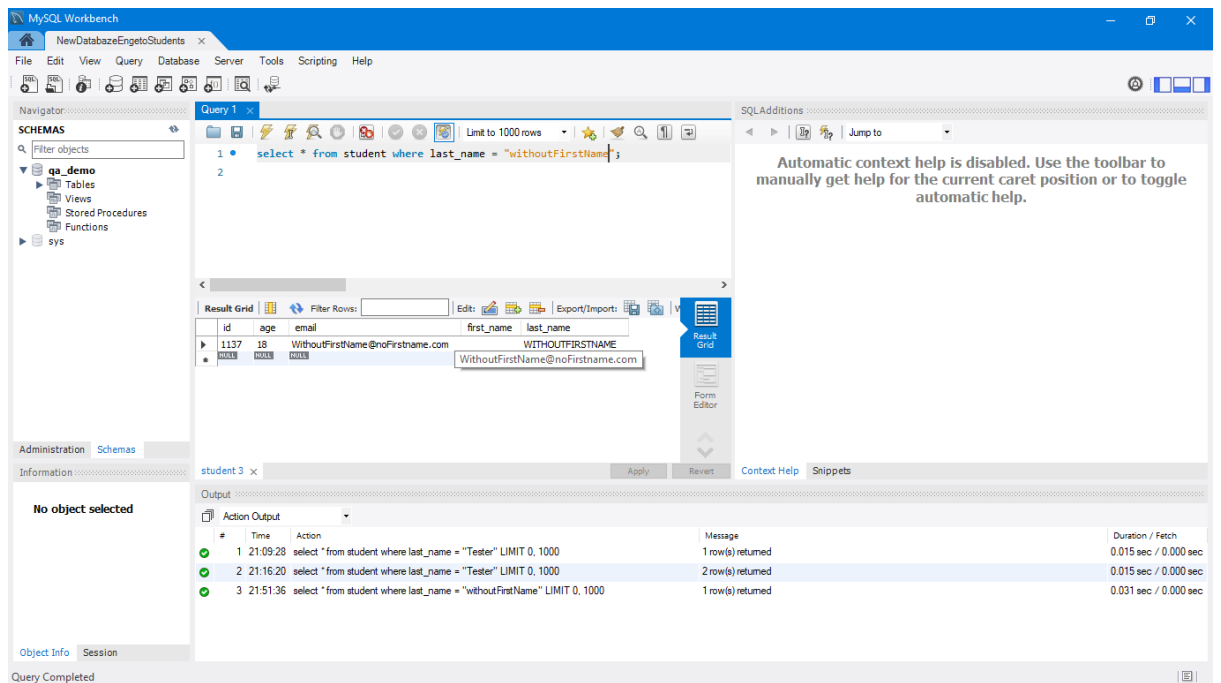
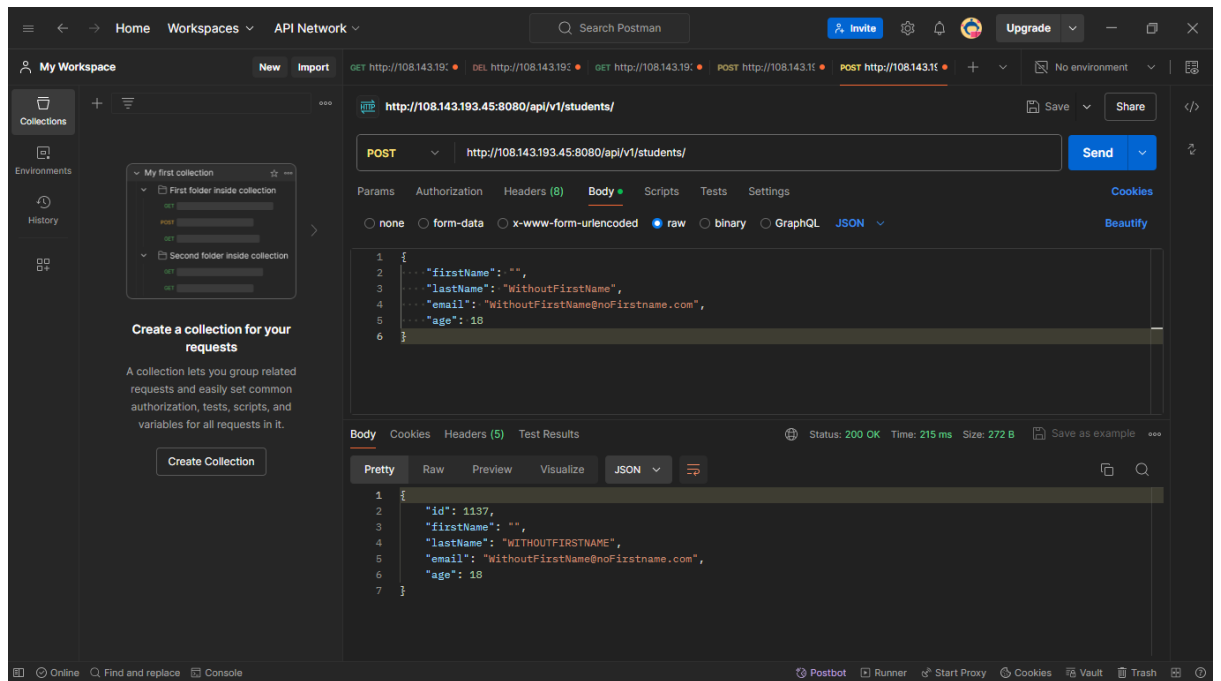
Status Code byl 200, očekávaný byl 201. Této chybě bylo již přiděleno ID 2.

Za chybu lze považovat možnost vytvořit duplicitu studenta, jen s tím rozdílem, že mu je přiděleno jiné ID. Této chybě bylo přiděleno ID3.

Očekávané výsledky: Očekával se výstup ve smyslu, že program vrátí chybu nebo hlášení, že není možné zadat všem atributům stejné hodnoty a tím vytvořit jednou již vytvořeného studenta, který se již nachází v databázi, akorát pod jiným ID číslem.
Očekával se status code 201.

3. Exekuce testovacího scénáře - výsledky

Abstrakt – Ověření metodou POST založení nového studenta bez zadání hodnoty do atributu "firstName" v databázi přes REST – API



Skutečné výsledky: Při provedení metody POST, při které byly vyplněny hodnoty u atributů lastName, email, age a nebyl vyplněn hodnotou atribut firstName, se založil do databáze nový student, kterému bylo vygenerováno ID 1137. Atributy firstName, lastName, email a age byly shodné jak s vyslaným požadavkem, tak se záznamem v databázi.

Hodnoty firstName, email a age byli shodné jak s vyslaným požadavkem, tak se záznamem v databázi.

Hodnota lastName se lišila s vyslaným požadavkem, kdy byl vyslán požadavek lastName: "WithoutFirstName" a výsledek byl lastName: "WITHOUTFIRSTNAME " . Této chybě bylo již přiděleno ID 1.

Status Code byl 200, očekávaný byl 201. Této chybě bylo již přiděleno ID 2.

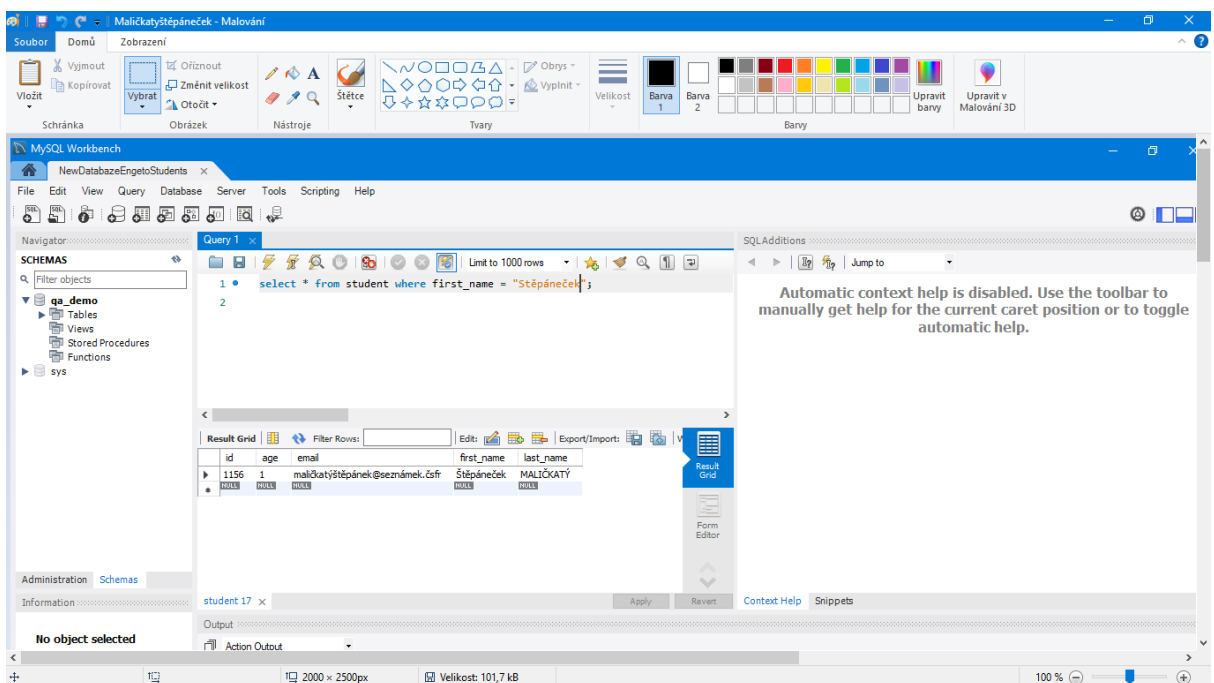
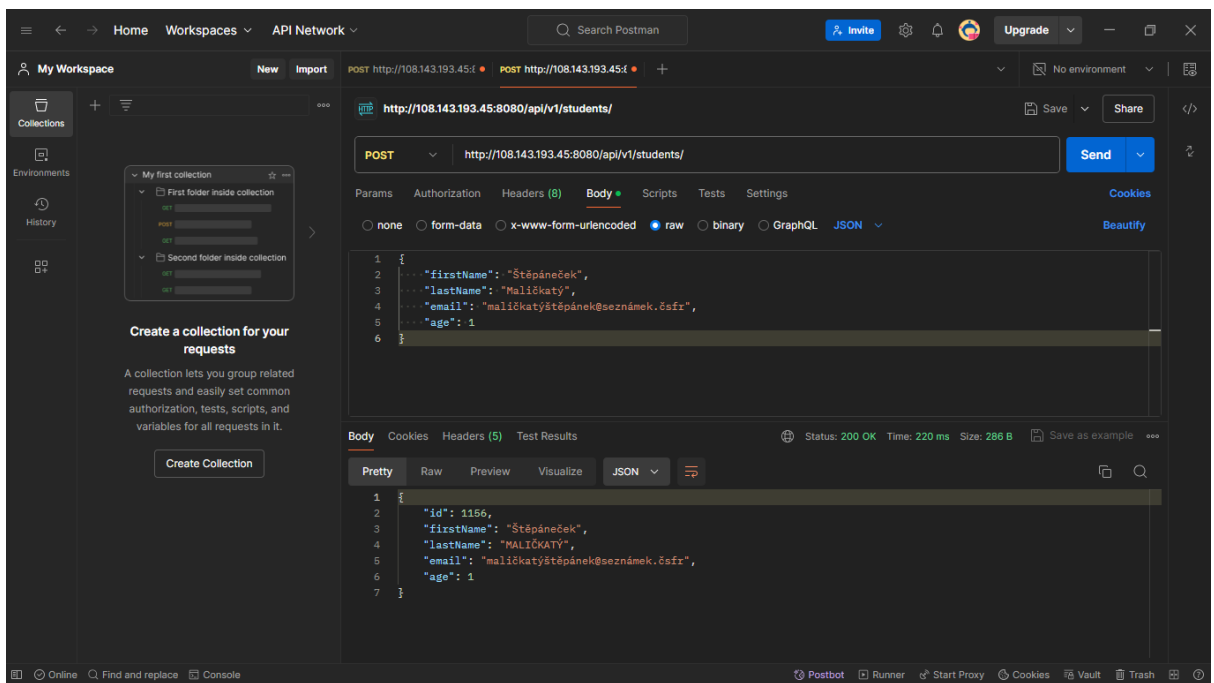
Zjistit požadavek, zda lze považovat za chybu to, že je možné založit nového studenta bez vyplněné hodnoty a atributu firstName. Tomuto je přiděleno ID 4.

Očekávané výsledky:

Očekávala se shoda jak v attributech, tak i v hodnotách .
Očekával se status code 201.

4.EXEKUCE TESTOVACÍHO SCÉNÁŘE

Abstrakt – Ověření metodou POST založení nového studenta v databázi přes REST – API, se zadáním hodnoty do atributu "firstName" s použitím diakritiky



Skutečné výsledky: Při provedení metody POST se založilo studentovi ID s číslem 1156.

Atributy firstName, lastName, email a age byly shodné jak s vyslaným požadavkem, tak se záznamem v databázi.

Hodnoty firstName, email a age byli shodné jak s vyslaným požadavkem, tak se záznamem v databázi.

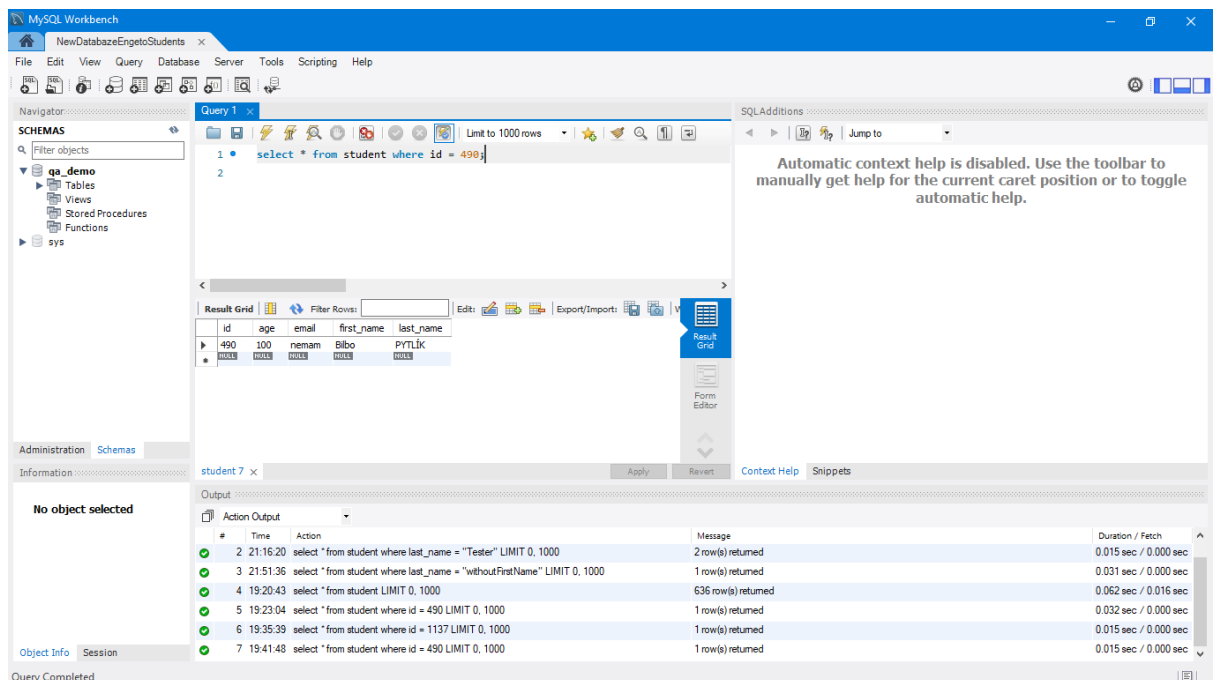
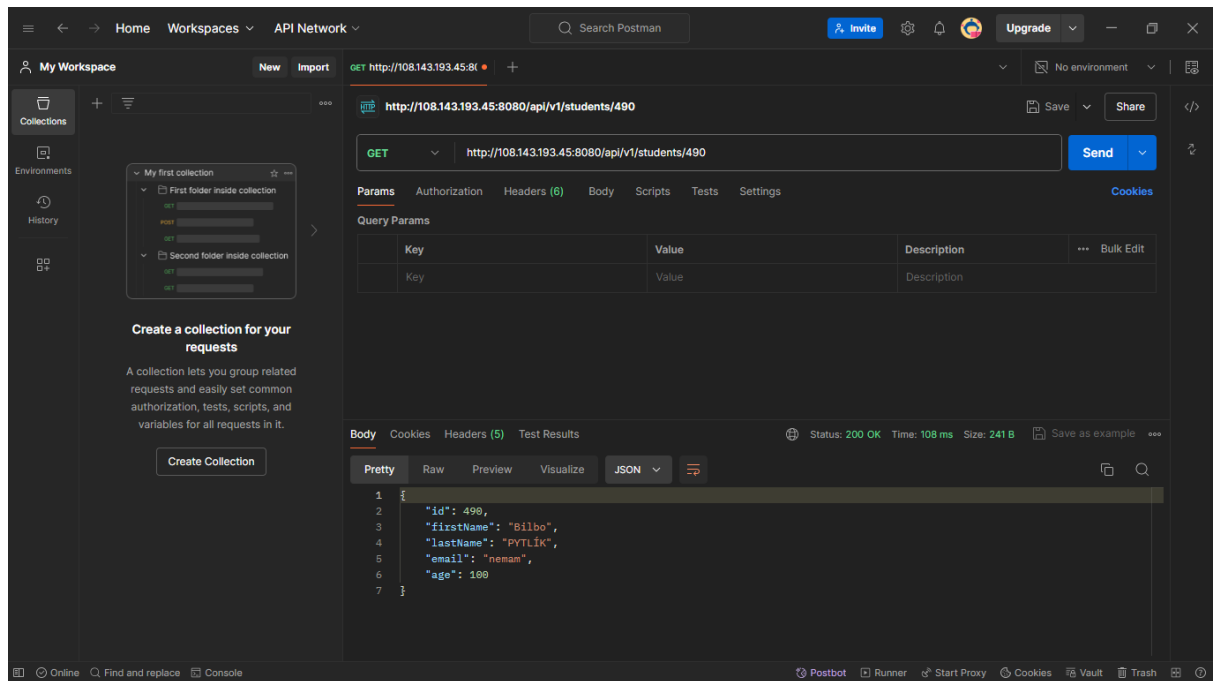
Hodnota lastName se lišila s vyslaným požadavkem, kdy byl vyslán požadavek lastName: "Maličkatý " a výsledek byl lastName: "MALIČKATÝ" .
Této chybě bylo již přiděleno ID 1.

Status Code byl 200, očekávaný byl 201. Této chybě bylo již přiděleno ID 2.

Očekávané výsledky: Očekávala se shoda jak v attributech, tak i v hodnotách – vyslaných v požadavcích = uloženými v databázi.
Očekávalo se náhodně vygenerované číslo ID studenta.
Očekával se status code 201.

2) METODA GET

5. Exekuce testovacího scénáře - výsledky



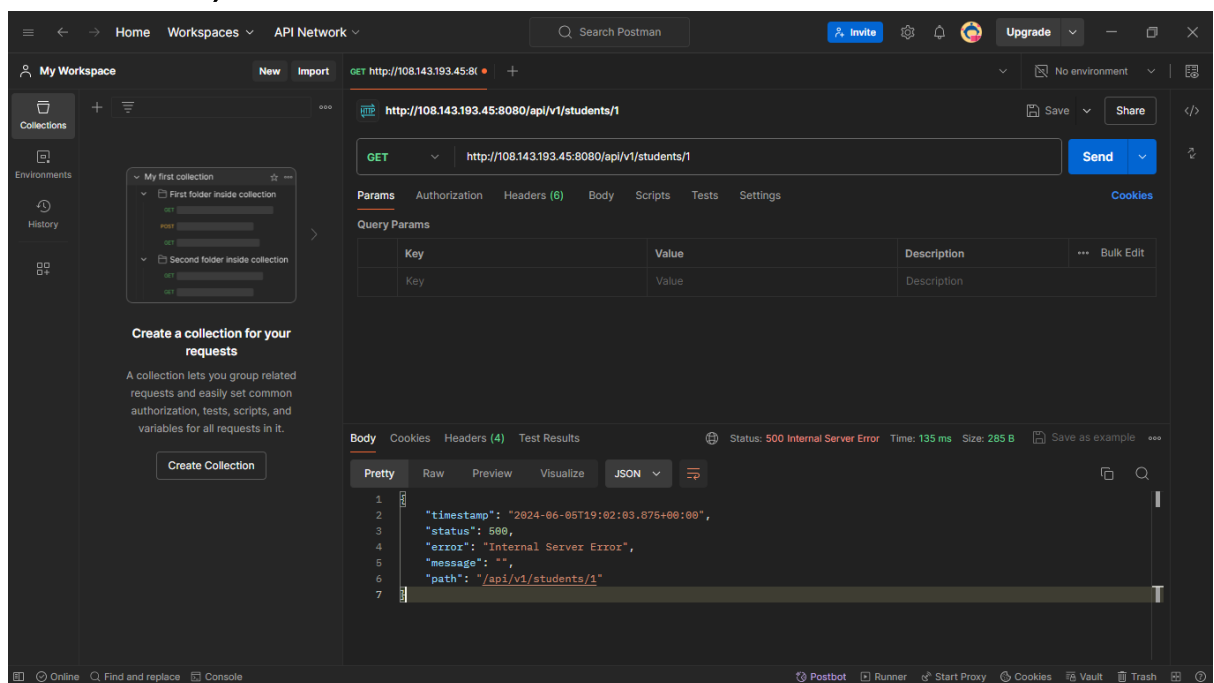
Skutečné výsledky: Při provedení metody GET s dotazem na studenta s ID490, byly hodnoty u atributů ID, firstName, lastName, email a age shodně zobrazené v programech Postman a My SQL Workbench. Atributy id, firtsName, lastName, email a age byly shodné jak s vyslaným požadavkem, tak se záznamem v databázi.

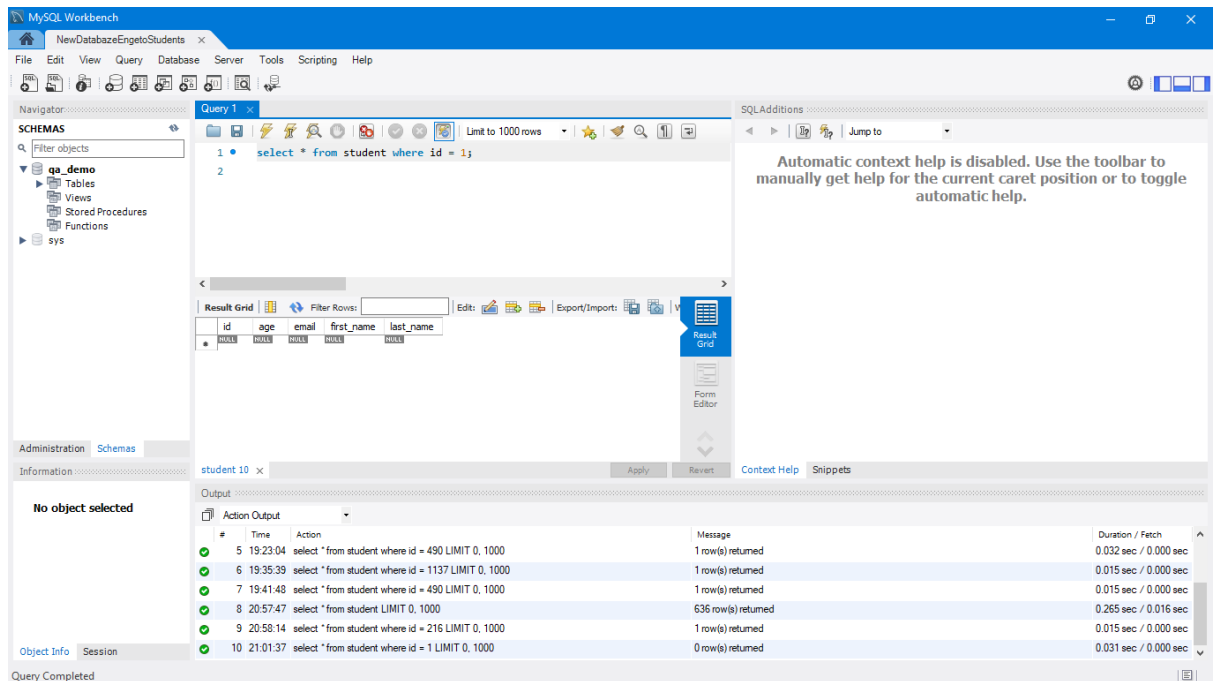
Status kód byl 200 OK.

Očekávané výsledky: Očekávala se shoda zobrazených dat v databázi zobrazenými v programu MYSQL Workbench s daty zobrazenými na request GET v programu Postman.
Očekával se status code 200 OK.

6.Exekuce testovacího scénáře - výsledky

Abstrakt – Získání dat z databáze metodou GET přes REST – API, zadané ID studenta je 1 (ID 1 – neexistující záznam s tímto ID v databázi)



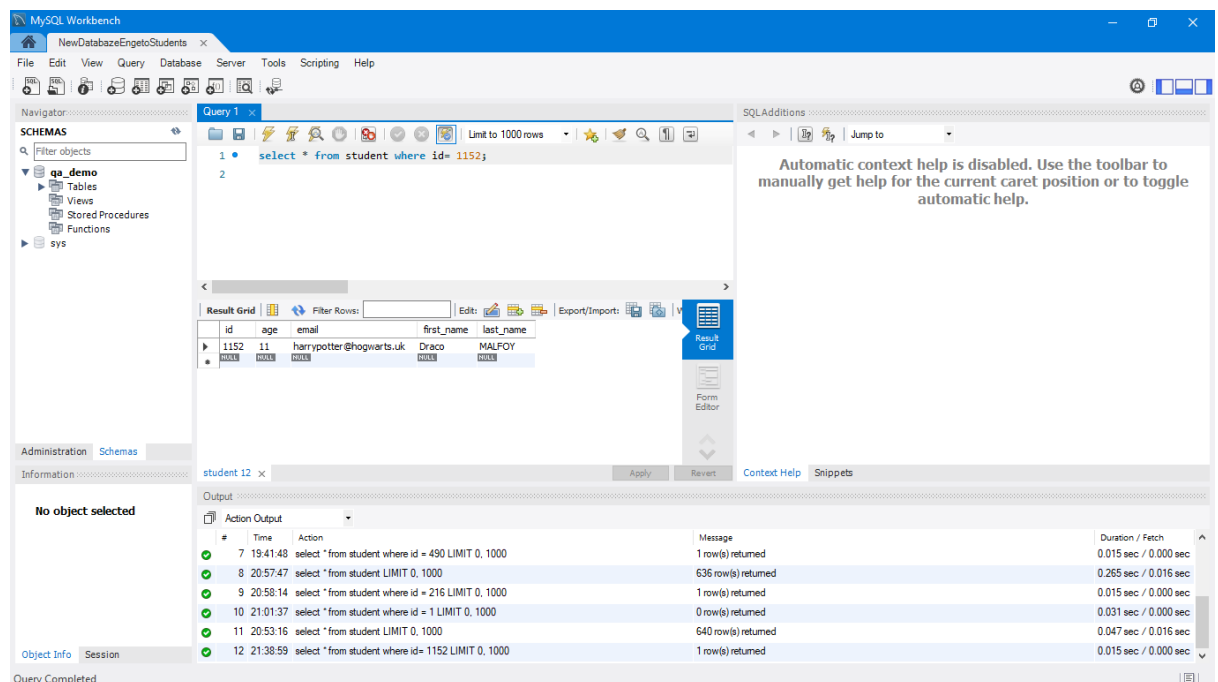
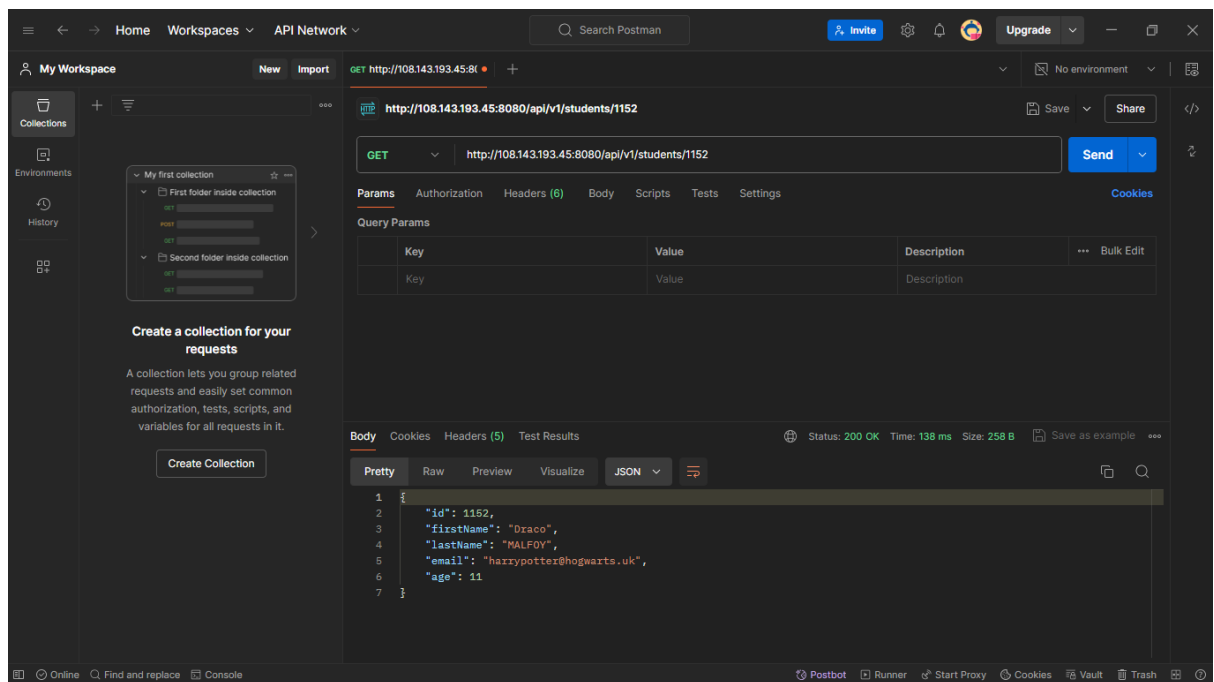


Skutečné výsledky: Při provedení metody GET s dotazem na neexistujícího studenta došlo k odpovědi se status codem 500 Internal Server Error.

Očekávané výsledky: Očekával se status code 404 not found, této chybě bylo přiřazeno ID 5.

7.Exekuce testovacího scénáře - výsledky

Abstrakt – Získání dat z databáze metodou GET přes REST – API, zadané ID studenta je 1152



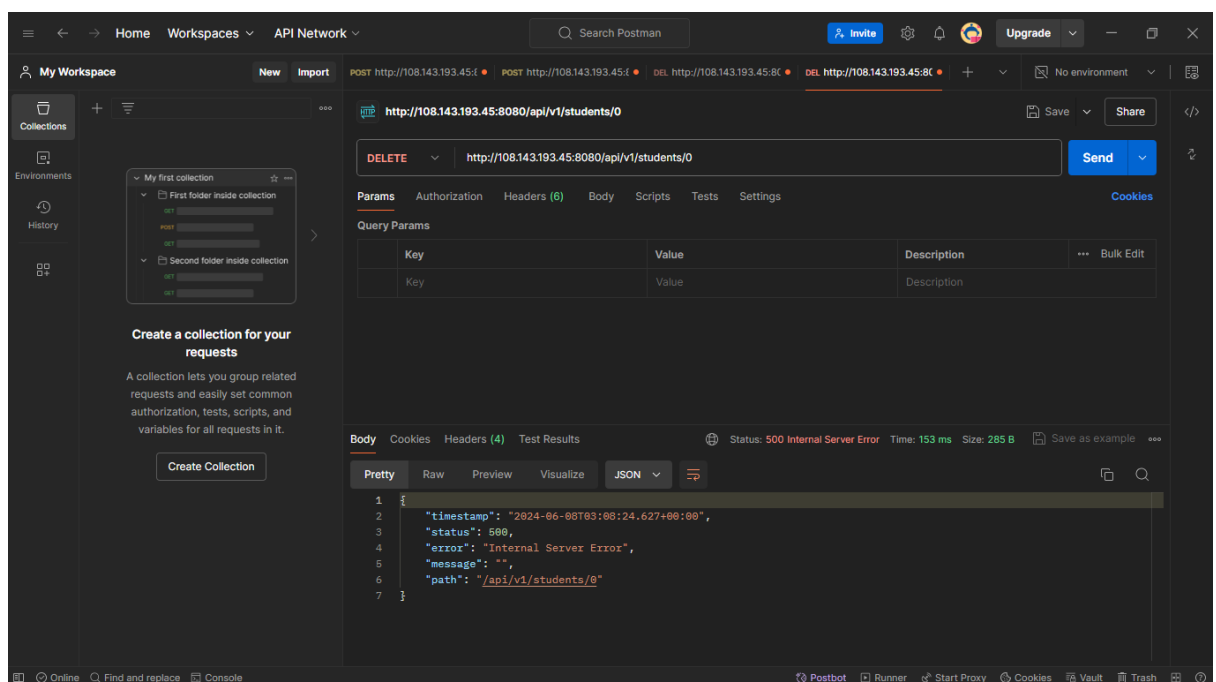
Skutečné výsledky: Při provedení metody GET s dotazem na studenta s ID1152, byly hodnoty u atributů ID, firstName, lastName, email a age shodně zobrazené v programech Postman a My SQL Workbench.

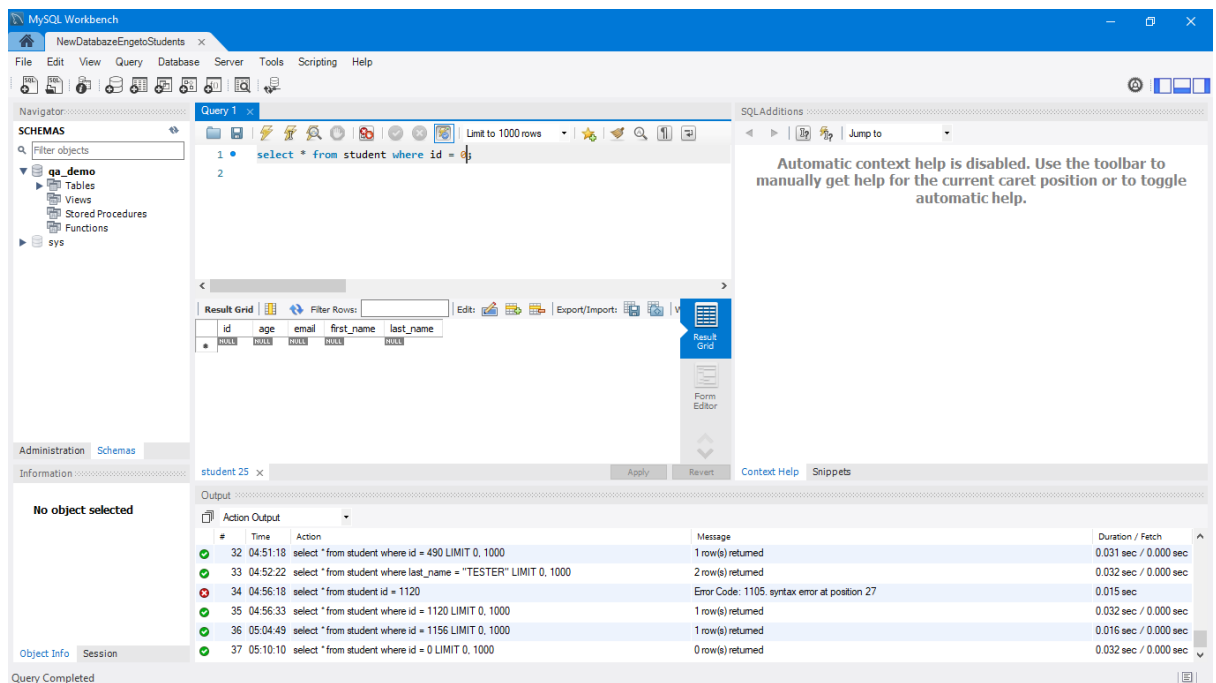
Atributy firstName, lastName, email a age byly shodné jak s vyslaným požadavkem, tak se záznamem v databázi.
Status kód byl 200 OK.

Očekávané výsledky: Očekávala se shoda zobrazených dat v databázi zobrazenými v programu MYSQL Workbench s daty zobrazenými na request GET v programu Postman.
Očekával se status code 200 OK.

3) METODA DELETE

8. Exekuce testovacího scénáře - výsledky
Abstrakt – Smazání studenta v databázi přes REST – API, student se zadaným ID 0 (“nula”) se nenachází v databázi





Skutečné výsledky: Při provedení metody DELETE pro smazání neexistujícího studenta s tímto ID byl status code 500 Internal Server Error.

Očekávané výsledky:

Očekávaný status code byl 404 Not Found. Této chybě bylo přiděleno ID 6.

9. Exekuce testovacího scénáře - výsledky
Abstrakt – Smazání studenta v databázi přes REST – API, student se zadaným ID 1120

The image shows two screenshots. The top screenshot is from Postman, showing a DELETE request to `http://108.143.193.45:8080/api/v1/students/1120` with a status of 200 OK. The bottom screenshot is from MySQL Workbench, showing the query `select * from student where id = 1120;` and its results in the 'Result Grid' tab. The results show 0 rows returned, indicating the student with ID 1120 was successfully deleted.

Postman Details:

- Method: DELETE
- URL: `http://108.143.193.45:8080/api/v1/students/1120`
- Status: 200 OK
- Time: 135 ms
- Size: 123 B

MySQL Workbench Details:

Query 1:

```
select * from student where id = 1120;
```

Result Grid:

id	age	email	first_name	last_name
----	-----	-------	------------	-----------

Output:

#	Time	Action	Message	Duration / Fetch
33	04:52:22	select * from student where last_name = "TESTER" LIMIT 0, 1000	2 row(s) returned	0.032 sec / 0.000 sec
34	04:56:18	select * from student where id = 1120	Error Code: 1105. syntax error at position 27	0.015 sec
35	04:56:33	select * from student where id = 1120 LIMIT 0, 1000	1 row(s) returned	0.032 sec / 0.000 sec
36	05:04:49	select * from student where id = 1156 LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
37	05:10:10	select * from student where id = 0 LIMIT 0, 1000	0 row(s) returned	0.032 sec / 0.000 sec
38	05:23:17	select * from student where id = 1120 LIMIT 0, 1000	0 row(s) returned	0.015 sec / 0.000 sec

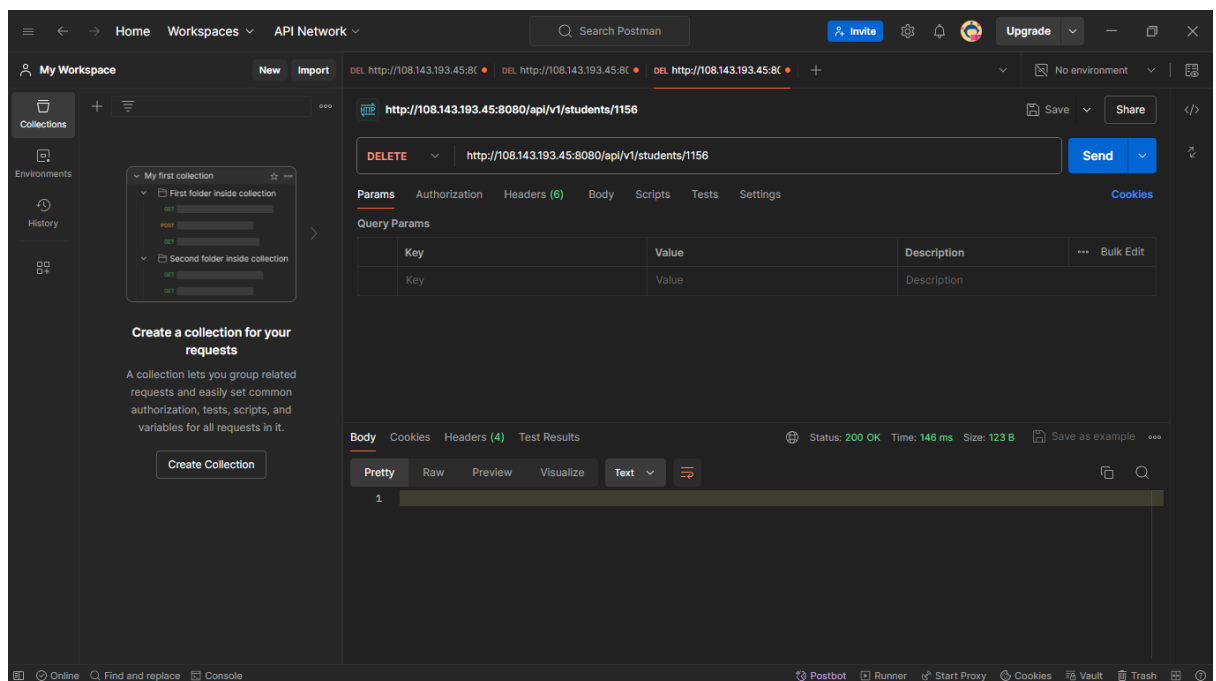
Skutečné výsledky: Při provedení metody DELETE pro smazání studenta s ID 1120 ID byl status code 200 OK. Došlo ke smazání celého záznamu.

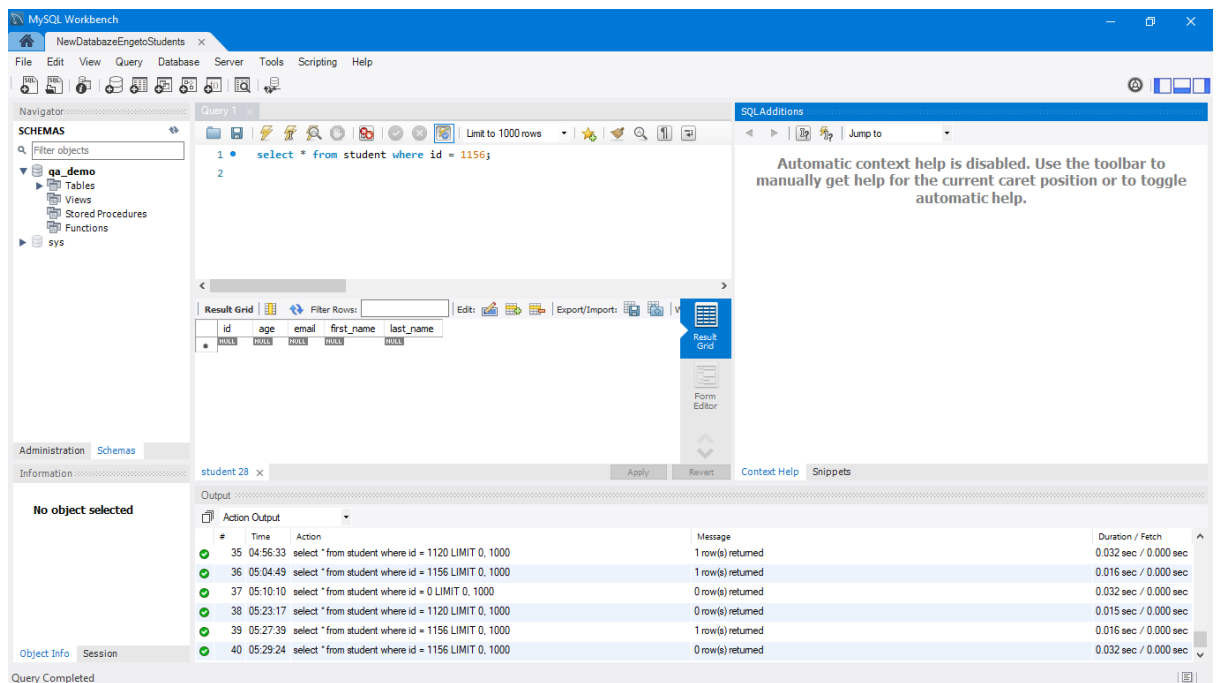
Očekávané výsledky:

Při provedení metody DELETE pro smazání studenta s ID 1120 ID byl očekáván status code 200 OK. Mělo dojít ke smazání celého záznamu.

10. Exekuce testovacího scénáře - výsledky

Abstrakt – Smazání studenta v databázi přes REST – API, student se zadaným ID 1156





Skutečné výsledky: Při provedení metody DELETE pro smazání studenta s ID 1156 ID byl status code 200 OK. Došlo ke smazání celého záznamu.

Očekávané výsledky:

Při provedení metody DELETE pro smazání studenta s ID 1156 ID byl očekáván status code 200 OK. Mělo dojít ke smazání celého záznamu.

BUG REPORT

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

1) METODA POST

1. Testovací scénář - byly zjištěny chyby

ID 1. Jedná se o špatné uložení hodnoty atributu lastName, kdy požadavek byl lastName: "Tester" a výstupem bylo lastName: " TESTER"

ID 2. Špatný status code. Očekávaný 201, skutečný 200. I přes špatný status code byl záznam v databázi vytvořen.

2. Testovací scénář - byly zjištěny chyby

ID 1. Jedná se o špatné uložení hodnoty atributu lastName, kdy požadavek byl lastName: "WithoutFirstName " a výstupem bylo lastName: " WITHOUTFIRSTNAME"

ID 2. Špatný status code. Očekávaný 201, skutečný 200. I přes špatný status code byl záznam v databázi vytvořen.

ID 3. Chyba v možnosti založit do databáze nového studenta, který je již v databázi aktuálně uložený. Tento student má jen jiné ID číslo, než má již aktuálně uložený.

3. Testovací scénář - byly zjištěny chyby

ID 1. Jedná se o špatné uložení hodnoty atributu lastName, kdy požadavek byl lastName: "WithoutFirstName " a výstupem bylo lastName: " WITHOUTFIRSTNAME"

ID 2. Špatný status code. Očekávaný 201, skutečný 200. I přes špatný status code byl záznam v databázi vytvořen.

ID 4. Zjistit požadavek, zda lze považovat za chybu to, že je možné založit nového studenta bez vyplněné hodnoty a atributu firstName.

4. SCÉNA TESTOVACÍ - byly zjištěny chyby

ID 1. Jedná se o špatné uložení hodnoty atributu firstName, kdy požadavek byl firstName: "Maličkatý" a výstupem bylo lastName: "MALIČKATÝ "

ID 2. Špatný status code. Očekávaný 201, skutečný 200. I přes špatný status code byl záznam v databázi vytvořen.

2) METODA GET

5. Testovací scénář - *nebyly zjištěny chyby*

6. Testovací scénář - *byly zjištěny chyby*

ID 5. Chybný Status code: 500. Dle očekávání měl být status code 404 not found.

7. Testovací scénář - *nebyly zjištěny chyby*

3) METODA DELETE

8. Testovací scénář - *byly zjištěny chyby*

ID 6. Špatný status code při pokusu o smazání neexistujícího studenta. Status Code byl 500 Internal Server Error, očekávaný byl 404 Not Found.

9. Testovací scénář - *nebyly zjištěny chyby*

10. Testovací scénář - *nebyly zjištěny chyby*