FINÁLNÍ PROJEKT
č.1

Autor: Ing. Daniel Jarnot
Datum: 14.4.2024
discord: amik4806

OBSAH

ZADÁNÍ

Čílem 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: 68q0llcbuvyijdt5mzq2 Password: pscale_pw_zxfenAXfSYx9loe9PCS1snuGKrFzw2J84BMlVZI96o
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.

Test case	1.0
Datum	14.4.2024
Abstract	Použití metody GET u existujícího studenta v databázi „student“
Verze	v01
Zdroj	Databáze „student“
Aplikace	Postman, MySQL Workbench
Kroky	1. Otevři aplikaci Postman a MySQL Workbench 2. V MySQL workbanch najdi existujícího studenta s id 210. 3. V aplikaci Postman vyber metodu GET a vlož adresu: http://108.143.193.45:8080/api/v1/students/**210** 4. Klikni na tlačítko „Send“ 5. V aplikaci Postman ověř, že údaje studenta s id 210 se shodují s údaji v databázi MySQL Workbench.

Sktutečný výsledek	{
	"id": 210,
	"firstName": "Anabela",
	"lastName": "LARSON",
	"email": "ana.larson@xxmail.com",
	"age": 25
	}
	
	Status: 200 OK

Očekáváný výsledek	{
	"id": 210,
	"firstName": "Anabela",
	"lastName": "LARSON",
	"email": "ana.larson@xxmail.com",
	"age": 25
	}
	
	Status: 200 OK

Test case	2.0
Datum	14.4.2024
Abstract	Použití metody GET u neexistujícího studenta v databázi „student“
Verze	v01
Zdroj	Databáze „student“
Aplikace	Postman, MySQL Workbench
Kroky	1. Otevři aplikaci Postman a MySQL Workbench 2. V MySQL workbanch zjisti, které číslo id není použito. V tomto případě id 177. 3. V aplikaci Postman vyber metodu GET a vlož adresu: http://108.143.193.45:8080/api/v1/students/**177** 4. Klikni na tlačítko „Send“ 5. V aplikaci Postman ověř, že nelze zobrazit údaje studenta pro id 177.

Sktutečný výsledek	{
	"timestamp": "2024-04-14T10:10:02.827+00:00",
	"status": 500,
	"error": "Internal Server Error",
	"message": "",
	"path": "/api/v1/students/177"
	}
	
	Status: 500 Internal Server Error

Očekáváný výsledek	{
	"timestamp": "2024-04-14T10:10:02.827+00:00",
	"status": 500,
	"error": "Internal Server Error",
	"message": "",
	"path": "/api/v1/students/177"
	}
	
	Status: 500 Internal Server Error

Test case	3.0
Datum	14.4.2024
Abstract	Vytvoření nového studenta pomocí metody POST.
Verze	v01
Zdroj	Databáze „student“
Aplikace	Postman, MySQL Workbench

Kroky	1. Otevři aplikaci Postman a MySQL Workbench
	2. V aplikaci Postman vyber metodu POST a vlož adresu:
	http://108.143.193.45:8080/api/v1/students/
	3.V aplikaci Postman do záložky Body vlož následující text:
	
	{
	"firstName": "Daniel",
	"lastName": "Jarnot",
	"age":33,
	"email": "dj90@gmail.com"
	}
	
	4. Klikni na tlačítko „Send“
	5. V aplikaci Postman zkontroluj, zda Status se rovná 200 OK.
	6. V aplikaci MySQL Workbench ověř, že se vytvořil nový záznam, dle zadaných údajů.

Sktutečný výsledek	{
	"id": 554,
	"firstName": "Daniel",
	"lastName": "JARNOT",
	"email": "dj90@gmail.com",
	"age": 33
	}
	
	Status: 200 OK

Očekáváný výsledek	{
	"id": 554,
	"firstName": "Daniel",
	"lastName": "JARNOT",
	"email": "dj90@gmail.com",
	"age": 33
	}
	
	Status: 200 OK

Test case	4.0
Datum	14.4.2024
Abstract	Vytvoření duplicitního studenta pomocí metody POST.
Verze	v01
Zdroj	Databáze „student“
Aplikace	Postman, MySQL Workbench

Kroky	1. Otevři aplikaci Postman a MySQL Workbench
	2. V aplikaci MySQL Workbench si vyber již existující záznam.
	3. V aplikaci Postman vyber metodu POST a vlož adresu:
	http://108.143.193.45:8080/api/v1/students/
	3.V aplikaci Postman do záložky Body vlož následující text. Tzn. Již existující údaje:
	
	{
	"firstName": "Daniel",
	"lastName": "Jarnot",
	"age":33,
	"email": "dj90@gmail.com"
	}
	
	4. Klikni na tlačítko „Send“
	5. V aplikaci Postman zkontroluj, zda Status se rovná 400 OK.
	6. V aplikaci MySQL Workbench ověř, že se nevytvořil duplicitní záznam.

Sktutečný výsledek	{
	"id": 556,
	"firstName": "Daniel",
	"lastName": "JARNOT",
	"email": "dj90@gmail.com",
	"age": 33
	}
	
	Status: 200 OK

Očekáváný výsledek	Status: 400 Bad Request

Test case	5.0
Datum	14.4.2024
Abstract	Psaní velkých a malých písmen
Verze	v01
Zdroj	Databáze „student“
Aplikace	Postman, MySQL Workbench

Kroky	1. Otevři aplikaci Postman a MySQL Workbench
	2. V aplikaci Postman vyber metodu POST a vlož adresu:
	http://108.143.193.45:8080/api/v1/students/
	3.V aplikaci Postman do záložky Body vlož následující text.
	
	{
	"firstName": "Michal",
	"lastName": "RYBÍZEK",
	"age": 18,
	"email": "mryb22@seznam.cz",
	}
	
	4. Klikni na tlačítko „Send“
	5. V aplikaci Postman zkontroluj, zda Status 200 OK.
	6. V aplikaci MySQL Workbench ověř, že se vytvořil nový údaj.

Sktutečný výsledek	{
	„id“: 557
	"firstName": "Michal",
	"lastName": "RYBÍZEK",
	"email": "mryb22@seznam.cz",
	"age": 18,
	}

Očekáváný výsledek	{
	„id“: 557
	"firstName": "Michal",
	"lastName": "Rybízek",
	"email": "mryb22@seznam.cz",
	"age": 18,
	}

Test case	6.0
Datum	14.4.2024
Abstract	Psaní čísel a písmen do pole „age“
Verze	v01
Zdroj	Databáze „student“
Aplikace	Postman, MySQL Workbench

Kroky	1. Otevři aplikaci Postman a MySQL Workbench
	2. V aplikaci Postman vyber metodu POST a vlož adresu:
	http://108.143.193.45:8080/api/v1/students/
	3.V aplikaci Postman do záložky Body vlož následující text.
	
	{
	"firstName": "Klaudie",
	"lastName": "Suchánková",
	"age": 41a,
	"email": "ksuch@seznam.cz",
	}
	
	3. Klikni na tlačítko „Send“
	4. V aplikaci Postman zkontroluj, zda Status se rovná 400 Bad Request.
	5. V aplikaci MySQL Workbench ověř, že se nevytvořil nový údaj.

Sktutečný výsledek	{
	"timestamp": "2024-04-14T15:39:11.881+00:00",
	"status": 400,
	"error": "Bad Request",
	"message": "",
	"path": "/api/v1/students/"
	}
	
	Status: 400 Bad Request

Očekáváný výsledek	{
	"timestamp": "2024-04-14T15:39:11.881+00:00",
	"status": 400,
	"error": "Bad Request",
	"message": "",
	"path": "/api/v1/students/"
	}
	
	Status: 400 Bad Request

Test case	7.0
Datum	14.4.2024
Abstract	Adresa emailu bez znaku: @
Verze	v01
Zdroj	Databáze „student“
Aplikace	Postman, MySQL Workbench

Kroky	1. Otevři aplikaci Postman a MySQL Workbench
	2. V aplikaci Postman vyber metodu POST a vlož adresu:
	http://108.143.193.45:8080/api/v1/students/
	3.V aplikaci Postman do záložky Body vlož následující text.
	
	{
	"firstName": "Martin",
	"lastName": "Jelito",
	"age": 49,
	"email": "mejelseznam.cz",
	}
	
	3. Klikni na tlačítko „Send“
	4. V aplikaci Postman zkontroluj, zda Status se rovná 400 Bad Request.
	5. V aplikaci MySQL Workbench ověř, že se nevytvořil nový údaj.

Sktutečný výsledek	{
	„id: 568,
	"firstName": "Martin",
	"lastName": "Jelito",
	"email": "mejelseznam.cz",
	"age": 49,
	}

Očekáváný výsledek	{
	"timestamp": "2024-04-14T15:39:11.881+00:00",
	"status": 400,
	"error": "Bad Request",
	"message": "",
	"path": "/api/v1/students/"
	}
	
	Status: 400 Bad Request

Test case	8.0
Datum	14.4.2024
Abstract	Psaní čísel a speciálních znaků do pole „firstName“ a „lastName“
Verze	v01
Zdroj	Databáze „student“
Aplikace	Postman, MySQL Workbench

Kroky	1. Otevři aplikaci Postman a MySQL Workbench
	2. V aplikaci Postman vyber metodu POST a vlož adresu:
	http://108.143.193.45:8080/api/v1/students/
	3.V aplikaci Postman do záložky Body vlož následující text.
	
	{
	"firstName": "kamil9@",
	"lastName": "7Jelito@7",
	"age": 89,
	"email": "kamjel7@seznam.cz",
	}
	
	3. Klikni na tlačítko „Send“
	4. V aplikaci Postman zkontroluj, zda Status se rovná 400 Bad Request.
	5. V aplikaci MySQL Workbench ověř, že se nevytvořil nový údaj.

Sktutečný výsledek	{
	„id: 563,
	"firstName": "kamil9@",
	"lastName": "7Jelito@7",
	"email": "kamjel7@seznam.cz",
	"age": 89,
	}
	
	Status: 200 OK

Očekáváný výsledek	{
	"timestamp": "2024-04-14T15:39:11.881+00:00",
	"status": 400,
	"error": "Bad Request",
	"message": "",
	"path": "/api/v1/students/"
	}
	
	Status: 400 Bad Request

Test case	9.0
Datum	14.4.2024
Abstract	Ostranění záznamu metodou DELETE
Verze	v01
Zdroj	Databáze „student“
Aplikace	Postman, MySQL Workbench
Kroky	1. Otevři aplikaci Postman a MySQL Workbench 2. V aplikaci MySQL Workbench najdi studenta s číslem id:491 3. V aplikaci Postman vyber metodu DELETE a vlož adresu: http://108.143.193.45:8080/api/v1/students/491 3. Klikni na tlačítko „Send“ 4. V aplikaci MySQL Workbench ověř, že se odstranil student s id: 491. 4. V aplikaci Postman zkontroluj, zda Status se rovná 200 OK.
Sktutečný výsledek	Status: 200 OK
Očekáváný výsledek	Status: 200 OK

Test case	10.0
Datum	14.4.2024
Abstract	Ostranění záznamu, který neexistuje
Verze	v01
Zdroj	Databáze „student“
Aplikace	Postman, MySQL Workbench
Kroky	1. Otevři aplikaci Postman a MySQL Workbench 2. V aplikaci MySQL Workbench v databázi „student“ si vyber číslo id, které nebylo použito. 3. V aplikaci Postman vyber metodu DELETE a vlož adresu: http://108.143.193.45:8080/api/v1/students/491 3. Klikni na tlačítko „Send“ 4. V aplikaci Postman zkontroluj, zda Status se rovná 500 Internal Error

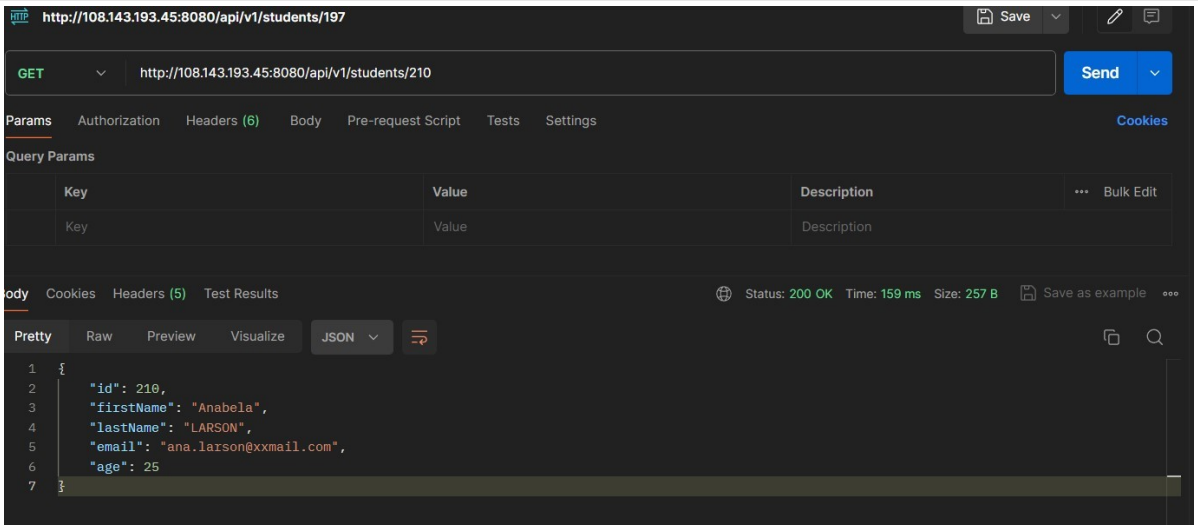
Sktutečný výsledek	{
	"timestamp": "2024-04-14T16:06:37.559+00:00",
	"status": 500,
	"error": "Internal Server Error",
	"message": "",
	"path": "/api/v1/students/491"
	}
	
	Status: 500 Internal Error

Očekáváný výsledek	{
	"timestamp": "2024-04-14T16:06:37.559+00:00",
	"status": 500,
	"error": "Internal Server Error",
	"message": "",
	"path": "/api/v1/students/491"
	}
	
	Status: 500 Internal Error

EXEKUCE TESTŮ

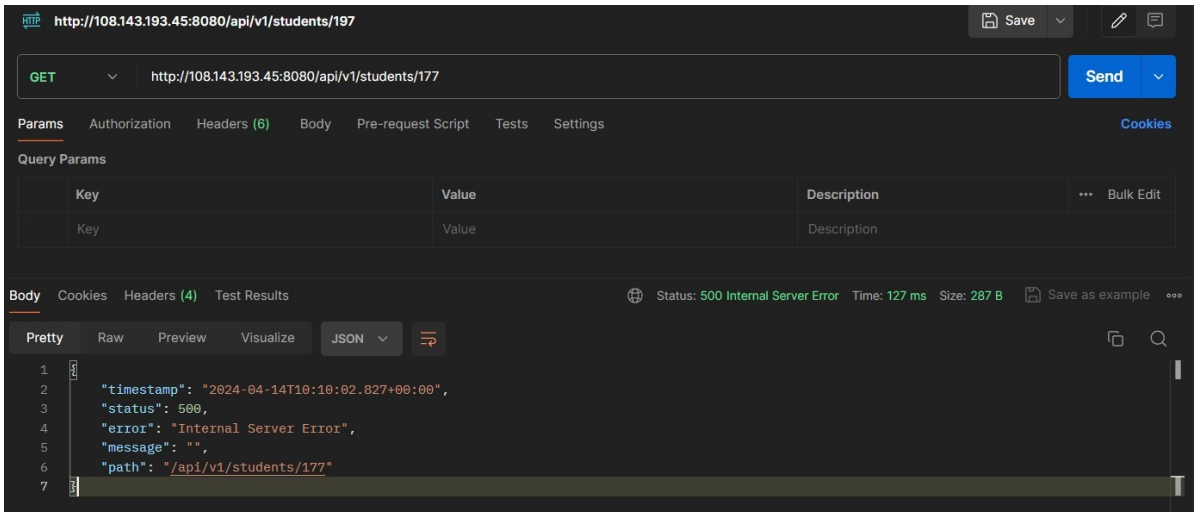
Testovací scénáře jsem provedl(a), přikládám výsledky testů.

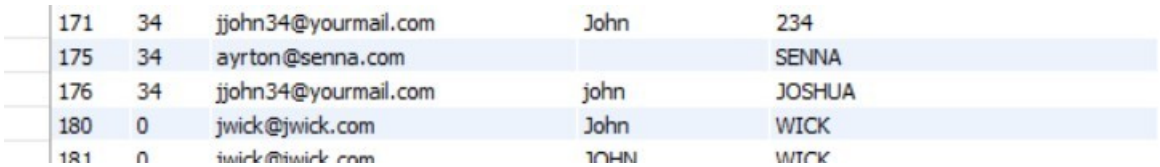
Test case 1.0: Použití metody GET u existujícího studenta v databázi „student“



209	34		John	JOSHUA
210	25	ana.larson@xxmail.com	Anabela	LARSON
211	34	jjohn34@yourmail.com		JOSHUA
213	34		John	JOSHUA

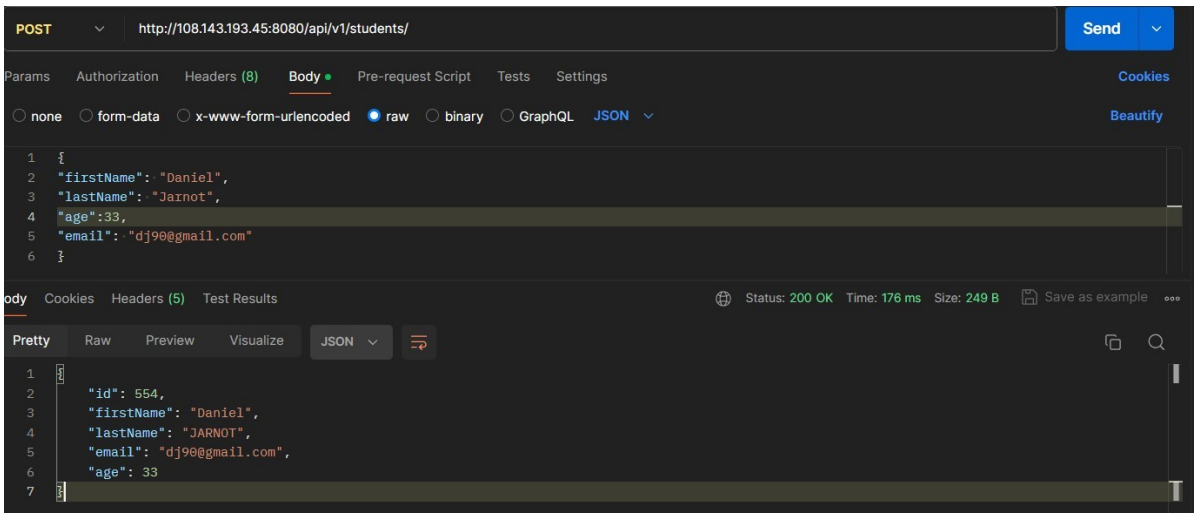
Test case 2.0: Použití metody GET u neexistujícího studenta v databázi „student“

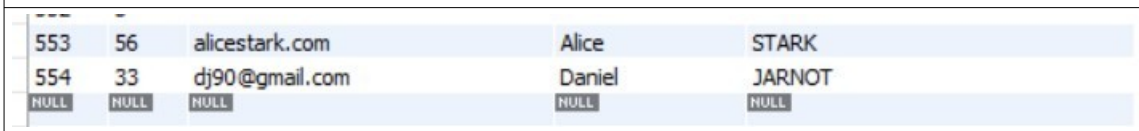




171	34	jjohn34@yourmail.com	John	234
175	34	ayrton@senna.com		SENNA
176	34	jjohn34@yourmail.com	john	JOSHUA
180	0	jwick@jwick.com	John	WICK
181	0	jwick@jwick.com	JOHN	WICK

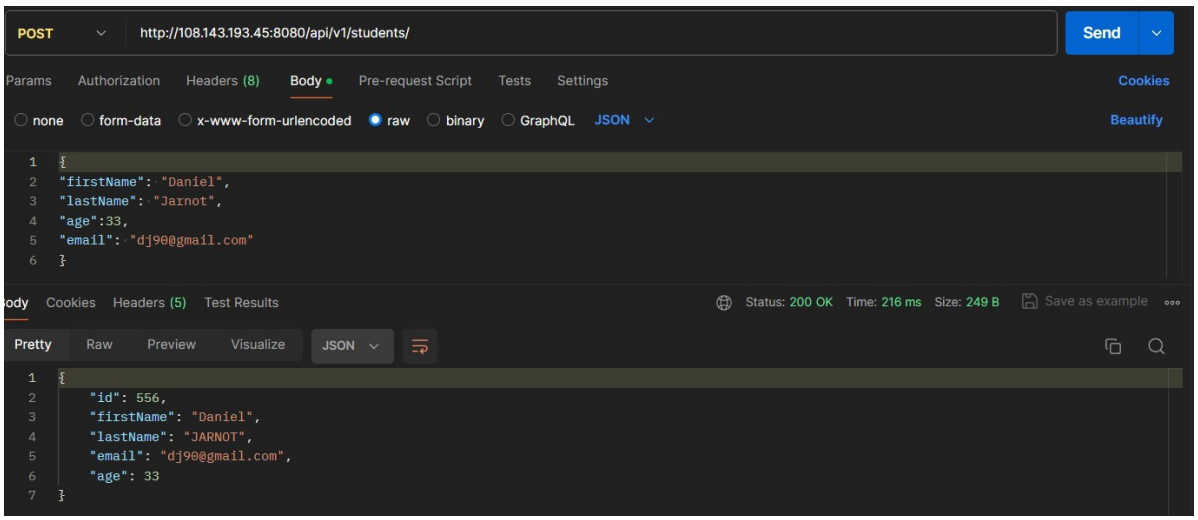
Test case 3.0: Vytvoření nového studenta pomocí metody POST.

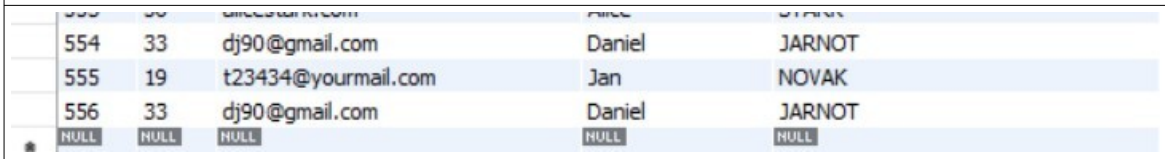




553	56	alicestark.com	Alice	STARK
554	33	dj90@gmail.com	Daniel	JARNOT
NULL	NULL	NULL	NULL	NULL

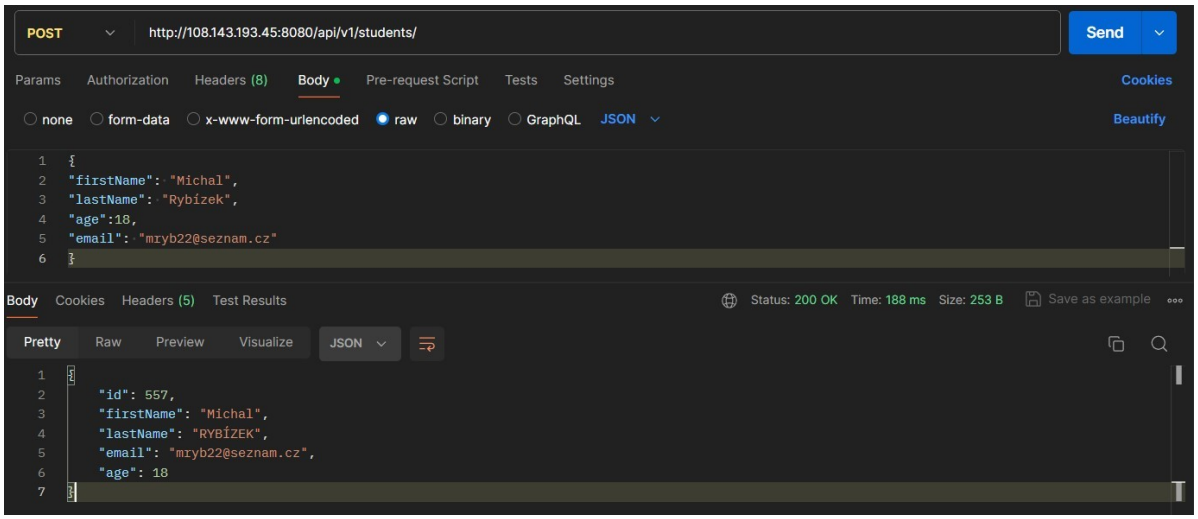
Test case 4.0: Vytvoření duplicitního studenta pomocí metody POST.





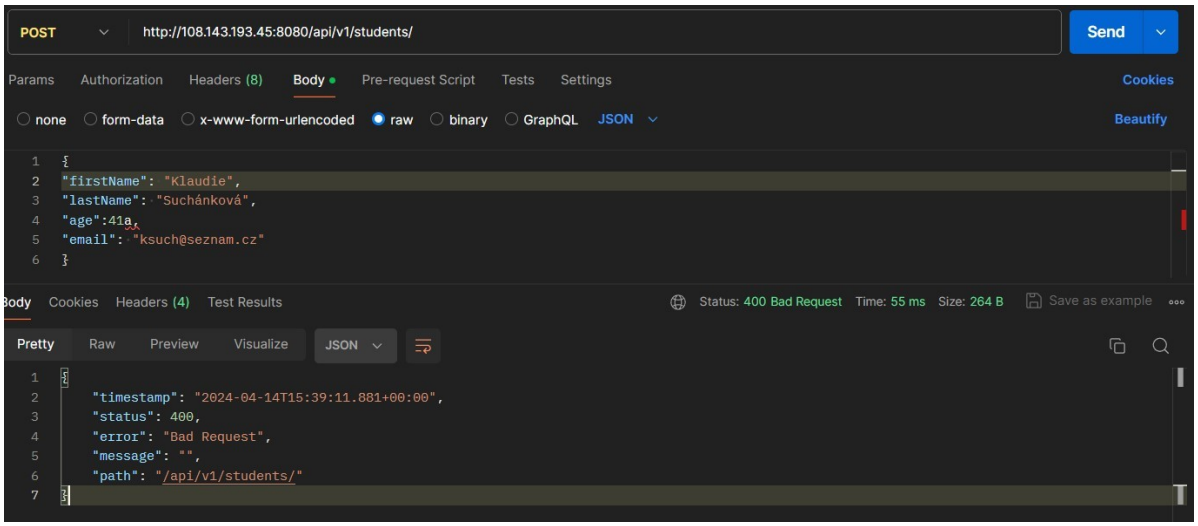
554	33	dj90@gmail.com	Daniel	JARNOT
555	19	t23434@yourmail.com	Jan	NOVAK
556	33	dj90@gmail.com	Daniel	JARNOT
NULL	NULL	NULL	NULL	NULL

Test case 5.0: Psaní velkých a malých písmen

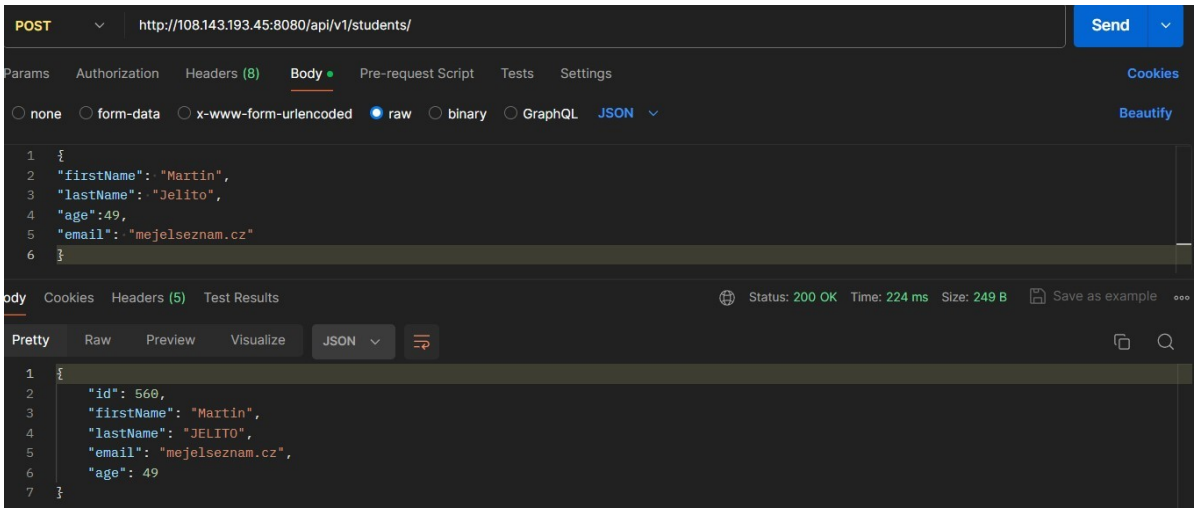


555	19	t23434@yourmail.com	Jan	NOVAK
556	33	dj90@gmail.com	Daniel	JARNOT
557	18	mryb22@seznam.cz	Michal	RYBÍZEK
NULL	NULL	NULL	NULL	NULL

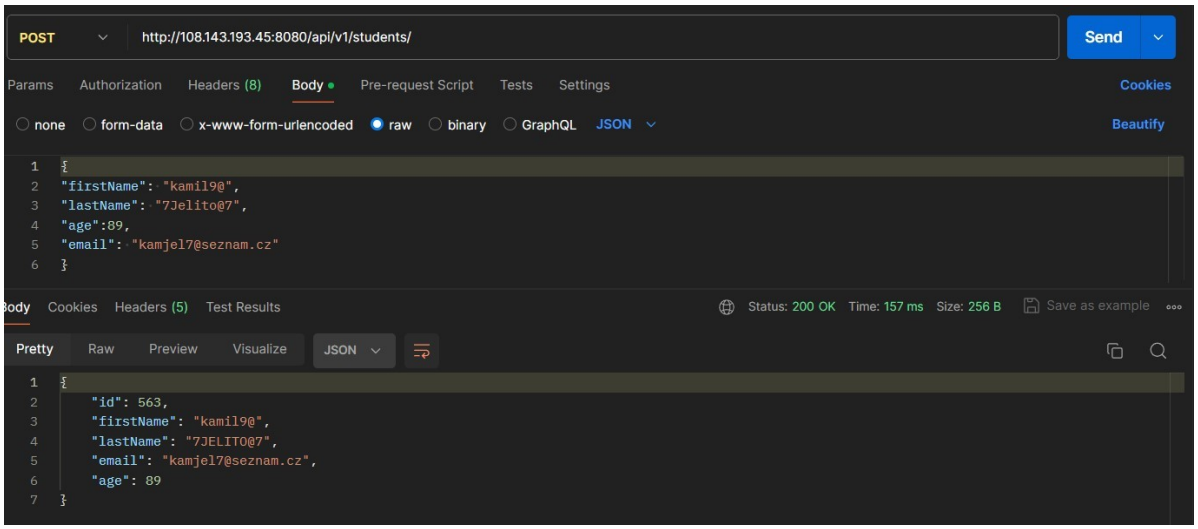
Test case 6.0: Psaní čísel a písmen do pole „age“

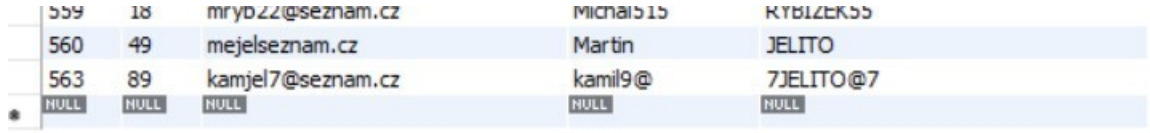


Test case 7.0: Adresa emailu bez znaku: @



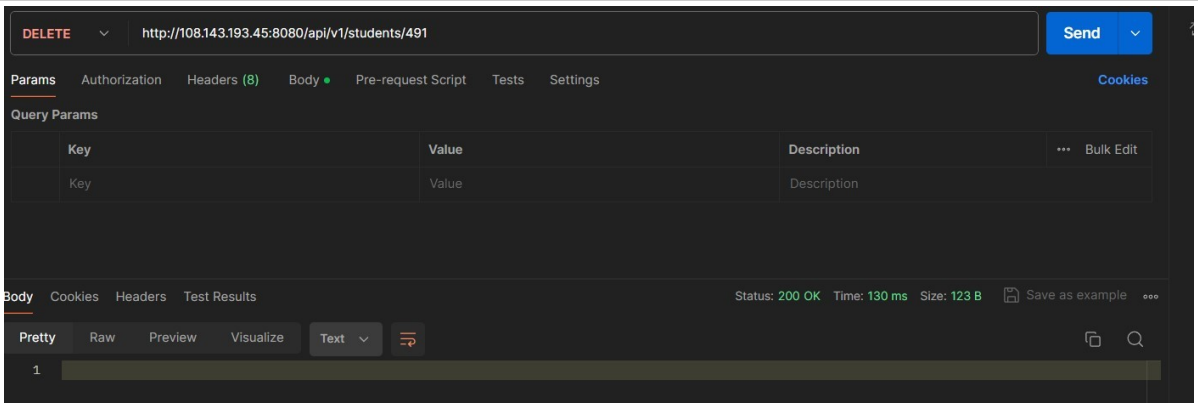
Test case 8.0: Psaní čísel a speciálních znaků do pole „firstName“ a „lastName“

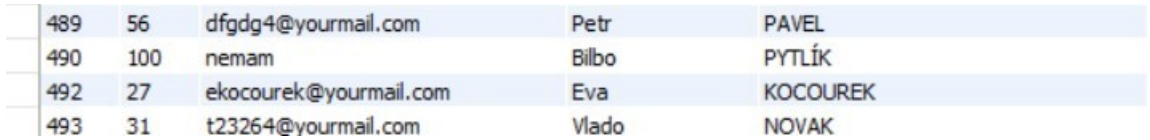




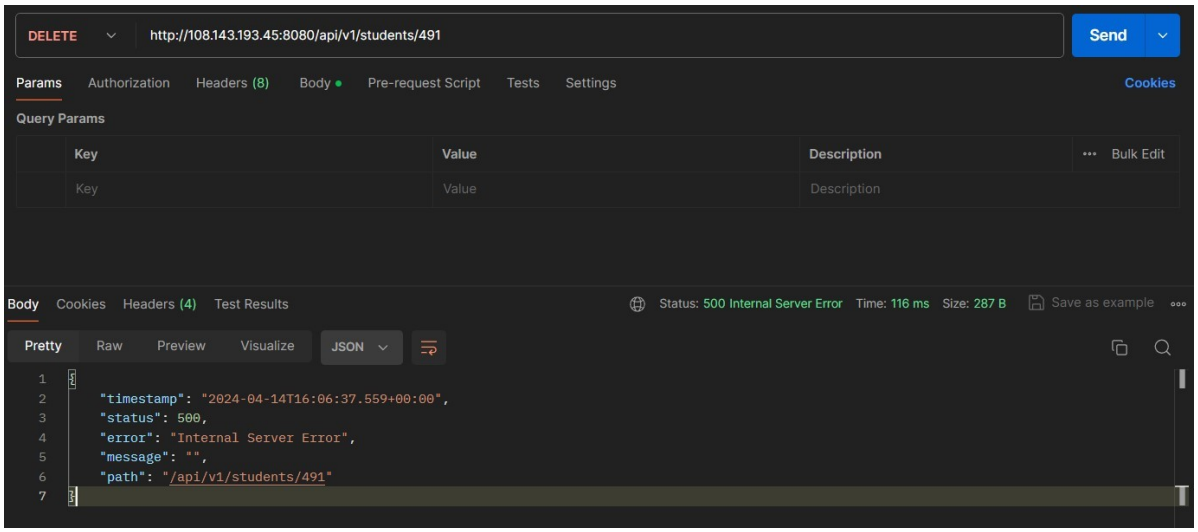
559	18	mryd22@seznam.cz	Michal515	RYBIZEK55
560	49	mejelseznam.cz	Martin	JELITO
563	89	kamjel7@seznam.cz	kamil9@	7JELITO@7
NULL	NULL	NULL	NULL	NULL

Test case 9.0: Ostranění záznamu metodou DELETE





489	56	dfgdg4@yourmail.com	Petr	PAVEL
490	100	nemam	Bilbo	PYTLÍK
492	27	ekocourek@yourmail.com	Eva	KOCOUREK
493	31	t23264@yourmail.com	Vlado	NOVAK

Test case 10.0: Ostranění záznamu, který neexistuje


BUG REPORT

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

Bug report	1.0
Test case	4.0
Datum	14.4.2024
Abstract	API umožní vytvořit duplicitní záznam s jiným id číslem.
Očekávaný výsledek	Mělo by být zakázáno vytváření záznamů se stejnou emailovou adresou.

Bug report	2.0
Test case	5.0
Datum	14.4.2024
Abstract	Pole „lastName“ – uloží se do databáze vždy velkými písmeny.
Očekávaný výsledek	Pokud do pole „lastName“ napíšu jméno malými písmeny, tak by se do databáze mělo uložit ve stejném formátu.

Bug report	3.0
Test case	7.0
Datum	14.4.2024
Abstract	Lze psát emailovou adresu bez znaku @.
Očekávaný výsledek	Pokud v emailové adrese bude chybět zavináč, pak by se měl spustit Status: 400 Bad Request

Bug report	4.0
Test case	8.0
Datum	14.4.2024
Abstract	Lze psát čísla a speciální znaky do pole „firstName“ a „lastName“
Očekávaný výsledek	Čísla a speciální znaky by měly být zakázány v poli „firstName“ a „lastName“. Měl by se spustit Status: 400 Bad Request