FINÁLNÍ PROJEKT
č.1

Autor: Kulykova Siuzanna Volodymyrivna
Datum: 11.03.2024

OBSAH

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.
REST rozhraní běží na adrese http://108.143.193.45:8080/api/v1/students/

Otestujte metody GET, POST a DELETE.

Platí, že:

1. metoda GET zobrazí data o existujícím studentovi
2. metoda POST založí záznam o studentovi
3. metoda DELETE smaže data o studentovi

TESTOVACÍ SCÉNÁŘE

Na základě uvedených testovacích scénářů jsem ověřila funkčnost aplikace.

Testovací scénář 1 Zkontrolují metodu GET, která zobrazí data o existujícím studentovi

Testovací případ 1

Název: Chování systému při zadání platného id studenta.

Testovací data: platné id studenta 352

Steps to reproduce:

1. Do requestu GET zadám http://108.143.193.45:8080/api/v1/students/352
2. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kód 200. Karta Body zobrazí JSON s údaji studenta.

Výsledek testu: **Test proběhl úspěšně.**

Testovací případ 2

Název: Chování systému při zadání neplatného id studenta

Testovací data: neplatné id studenta 888

Steps to reproduce:

1. Do requestu GET zadám http://108.143.193.45:8080/api/v1/students/888
2. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kod 404.

Výsledek testu: **Test proběhl neúspěšně.**

Testovací případ 3

Název: Získání dat všech studentů

Testovací data:

Steps to reproduce:

1. Do requestu GET zadám http://108.143.193.45:8080/api/v1/students/
2. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kód 200. Karta Body vydá JSON všech studentů z databáze.

Výsledek testu: **Test proběhl úspěšně.**

Testovací případ 4

Název: Chování systému při zadání mezer na místo id studenta

Testovací data: mezera

Steps to reproduce:

1. Do requestu GET zadám http://108.143.193.45:8080/api/v1/students/
2. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kód 400.

Výsledek testu: **Test proběhl neúspěšně.**

Testovací scénář 2 Zkontrolují metodu POST, která založí záznam o studentovi

Testovací případ 5

Název: Založení dat studenta, když je zadáno údaje do všech vstupů.

Testovací data:

"firstName":" Alexa",
"lastName": " Doležal",
"email": "a_doležal@gmail.com",

"age": 30

Steps to reproduce:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kód 200. Karta Body obsahuje JSON studenta, struktura kterého:
{
"id":____,
"firstName":" Alexa",
"lastName": " Doležal",
"email": "a_doležal@gmail.com",
"age": 30
}
Výsledek testu: Test proběhl neúspěšně.

Testovací případ 6

Název: Odmítnutí vložení údajů do databáze, když není zadáno pouze firstName

Testovací data:
"lastName": "Doležal",
"email": "a_doležal@gmail.com",
"age": 30

Steps to reproduce:
1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek: Vrací se stavový kód 400.

Výsledek testu: Test proběhl neúspěšně.

Testovací případ 7

Název: Odmítnutí vložení údajů do databáze, když není zadáno pouze lastName

Testovací data:
"firstName":"Alexa",

"email": "a_doležal@gmail.com",
"age": 30

Steps to reproduce:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kód 400.

Výsledek testu: Test proběhl neúspěšně.

Testovací případ 8

Název: Odmítnutí vložení údajů do databáze, když není zadán pouze email

Testovací data:
"firstName":"Alexa",
"lastName": "Doležal",
"age": 30

Steps to reproduce:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kód 400.

Výsledek testu: Test proběhl neúspěšně.

Testovací případ 9

Název: Odmítnutí vložení údajů do databáze, když není zadán věk

Testovací data:
"firstName":"Alexa",
"lastName": "Doležal",
"email": "a_doležal@gmail.com"

Steps to reproduce:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek: Vrací se stavový kód 400.

Testovací případ 10

Název: Odmítnutí vložení údajů do databáze, když nejsou zadány žádné údaje

Testovací data:

Steps to reproduce:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw ne zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kód 400.

Výsledek testu: Test proběhl neúspěšně.

Testovací případ 11

Název: Odmítnutí vložení údajů do databáze, když je zadáno speciální symboly do políčka firstName

Testovací data:

"firstName":"@#$%^&*()",
"lastName": "Doležal",
"email": "a_doležal@gmail.com",
"age": 30

Steps to reproduce:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kód 400.

Výsledek testu: Test proběhl neúspěšně.

Testovací případ 12

Název: Odmítnutí vložení údajů do databáze, když je zadáno mezery do políčka firstName
Testovací data:

"firstName":" ",
"lastName": "Doležal",
"email": "a_doležal@gmail.com",
"age": 30

Steps to reproduce:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek: Vrací se stavový kód 400.

Výsledek testu: Test proběhl neúspěšně.

Testovací případ 13

Název: Odmítnutí vložení údajů do databáze, když je zadáno číslice do políčka firstName

Testovací data:

"firstName":"12345",
"lastName": "Doležal",
"email": "a_doležal@gmail.com",
"age": 30

Steps to reproduce:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kód 400.

Výsledek testu: Test proběhl neúspěšně.
Testovací případ 14

Název: Odmítnutí vložení údajů do databáze, když je zadáno údaj délkou 256 symbolu do políčka firstName

Testovací data:

"firstName":"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",
"lastName": "Doležal",
"email": "a_doležal@gmail.com",
"age": 30

Steps to reproduce:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek:

Vrací se stavový kód 400.

Výsledek testu: Test proběhl neúspěšně.

Testovací scénář 3 Zkontrolují metodu DELETE, která smaže data o studentovi

Testovací případ 15

Název: Chování systému při smazání data platného id studenta

Testovací data: platné id studenta na moment provedení testu

Steps to reproduce:

1. Do requestu DELETE zadám http://108.143.193.45:8080/api/v1/students/"platné id studenta na moment provedení testu"
2. Zmačknu tlačítko Send

Očekávaný výsledek:
Vrací se stavový kód 200. Parametry JSON Body jsou prázdné.

Výsledek testu: Test proběhl úspěšně.

Testovací případ 16

Název: Chování systému při pokusu smazat data neplatného id studenta

Testovací data: neplatné id studenta 888

Steps to reproduce:

1. Do requestu DELETE zadám http://108.143.193.45:8080/api/v1/students/888
2. Zmačknu tlačítko Send

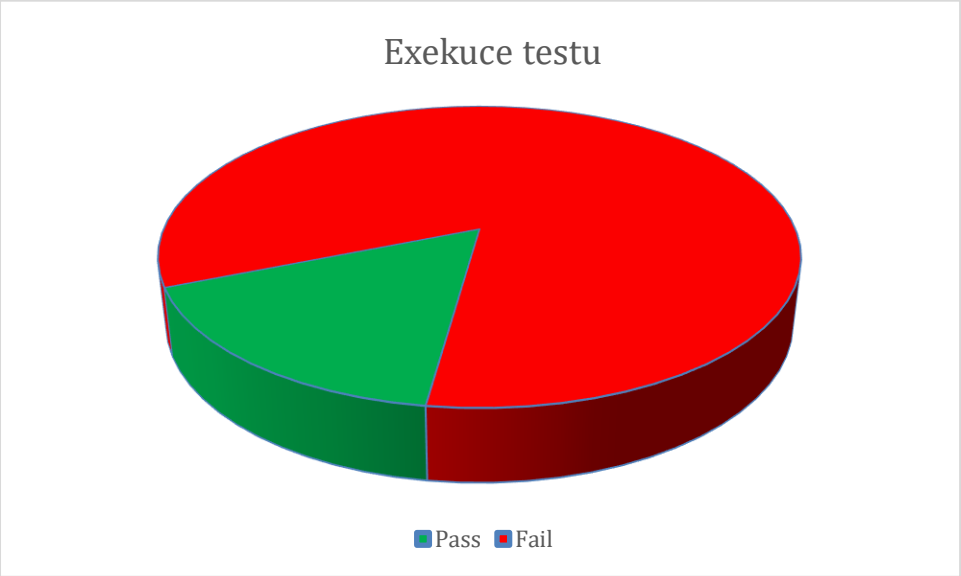
Očekávaný výsledek: Vrací se stavový kód 404.

Výsledek testu: **Test proběhl neúspěšně.**

EXEKUCE TESTŮ

Testovací scénáře jsem provedla, přikládám výsledky testů.

Bylo provedeno 16 testů, z nichž 13 bylo neúspěšných a 3 úspěšných.



K chybám můžou být přiřazeny následující druhy závažnosti (severity):

- **Blocker**: Chyba, která brání dalšímu testování nebo použití systému. Musí být vyřešena okamžitě.
- **Critical**: Chyba, která způsobuje selhání hlavních funkcí aplikace.
- **Major**: Chyba, která významně ovlivňuje funkčnost, ale neblokuje celkovou práci.
- **Minor**: Chyba, která má malý dopad na funkcionalitu.
- **Trivial**: Chyba, která je kosmetická nebo má minimální vliv na uživatele.

BUG REPORTS

Na základě provedených scénářů jsem objevila uvedené chyby aplikace.

bug	probability	severity	test case number	mitigace
1. Vrací se kód 500 při zadání neplatného id studenta	Deterministická	**Major**	2	Místo toho by měl být vrácen kód 404 (Not Found). Na straně serveru je třeba přidat validaci, která zkontroluje formát a strukturu ID před provedením dotazu do databáze. Poskytování jasných a specifických chybových zpráv pomůže uživatelům i vývojářům lépe porozumět problému a rychleji jej vyřešit.
2. Vrací se kód 500 při zadání mezer na místo id studenta	Deterministická	**Major**	4	Přidat validaci, která zkontroluje, zda ID neobsahuje pouze mezery. Pokud jsou zjištěny mezery, vrátí kód 400.
3. Příjmení se v databázi ukládá velkými písmeny.	Deterministická	**Minor**	5	Ujisti se, že v SQL dotazu není žádná funkce, která by převáděla text na velká písmena a opravit to.
4. Vrací se kód 500 při vložení údajů do databáze, když není zadáno pouze firstName	Deterministická	**Major**	6	Přidat kontrolu, která zajistí, že všechny požadované údaje jsou přítomny před vložením do databáze. Pokud chybí firstName nebo jakýkoliv jiný povinný vstup, vrátí kód 400 (Bad Request).
5. Vrací se kód 500 při vložení údajů do databáze, když není zadáno pouze lastName	Deterministická	**Major**	7	Přidat kontrolu, která zajistí, že všechny požadované údaje jsou přítomny před vložením do databáze. Pokud chybí lastName nebo jakýkoliv jiný povinný vstup, vrátí kód 400 (Bad Request).
6. Vrací se kód 500 při vložení údajů do databáze, když není zadán pouze email	Deterministická	**Major**	8	Přidat kontrolu, která zajistí, že všechny požadované údaje jsou přítomny před vložením do databáze. Pokud chybí email nebo jakýkoliv jiný povinný vstup, vrátí kód 400 (Bad Request).
7. Vrací se kód 500 při vložení údajů do databáze, když není zadán věk	Deterministická	**Major**	9	Přidat kontrolu, která zajistí, že všechny požadované údaje jsou přítomny před vložením do databáze. Pokud chybí věk nebo jakýkoliv jiný povinný vstup, vrátí kód 400 (Bad Request).

8. Vrací se kód 500 při vložení údajů do databáze, když nejsou zadány žádné údaje	Deterministická	**Major**	10	Přidat kontrolu, která zajistí, že všechny požadované údaje jsou přítomny před vložením do databáze. Pokud chybí jakýkoliv jiný povinný vstup, vrátí kód 400 (Bad Request).
9. Vložení údajů do databáze, když je zadáno speciální symboly do políčka firstName	Deterministická	**Major**	11	V SQL je nutné omezit ukládání čísel mezer a spec symbolu do polí s textovým datovým typem pomocí kontrolních omezení (constraints) při vytváření tabulek. Konkrétně lze použít CHECK constraints, které zajistí, že do daného pole budou ukládána pouze data, která splňují určitá kritéria.
10. Vložení údajů do databáze, když je zadáno mezery do políčka firstName	Deterministická	**Major**	12	Přidat kontrolu, která zajistí, že všechny požadované údaje jsou přítomny před vložením do databáze. Pokud chybí jakýkoliv jiný povinný vstup, vrátí kód 400 (Bad Request).
11. Vložení údajů do databáze, když je zadáno číslice do políčka firstName	Deterministická	**Major**	13	V SQL je nutné omezit ukládání čísel mezer a spec symbolu do polí s textovým datovým typem pomocí kontrolních omezení (constraints) při vytváření tabulek. Konkrétně lze použít CHECK constraints, které zajistí, že do daného pole budou ukládána pouze data, která splňují určitá kritéria. Omezit maximální délku pro textové sloupce v SQL tabulek při pomocí atributu VARCHAR. Prakticky je důležité vyřešit a stanovit jasná kritéria pro vstupní informace. Bude se uvažovat o zavedení dvojitých jmen nebo příjmení apod? Podle toho stanovit maximální počet znaků, povolené znaky nebo mezery. Stejně tak pro všechna vstupní políčka.
12. Vložení údajů do databáze, když je zadáno údaj délkou 256 symbolu do políčka firstName.	Deterministická	**Minor**	14	
13. Vrací se kód 500 při pokusu smazat data neplatného id studenta	Deterministická	**Major**	16	Namísto toho by měl být vrácen kód 404 (Not Found). Na straně serveru je třeba přidat validaci, která zkontroluje formát a strukturu ID před provedením dotazu do databáze.

Bug report 1

Název: Vrací se kód 500 při zadání neplatného id studenta
Závažnost: Major

Kroky k reprodukci chyby:

1. Do requestu GET zadám http://108.143.193.45:8080/api/v1/students/888
2. Zmačknu tlačítko Send

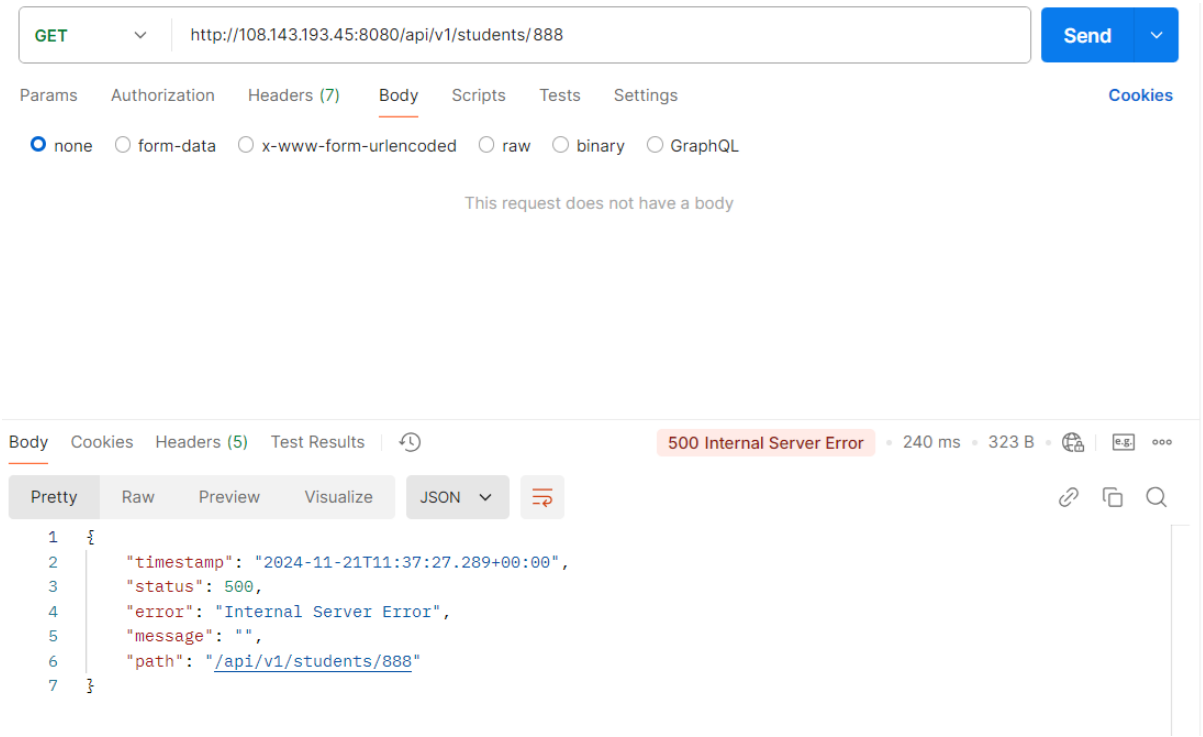
Očekávaný výsledek:

Vrací se stavový kod 404.

Skutečný výsledek:

Vrací se kód 500

Příloha:



Bug report 2

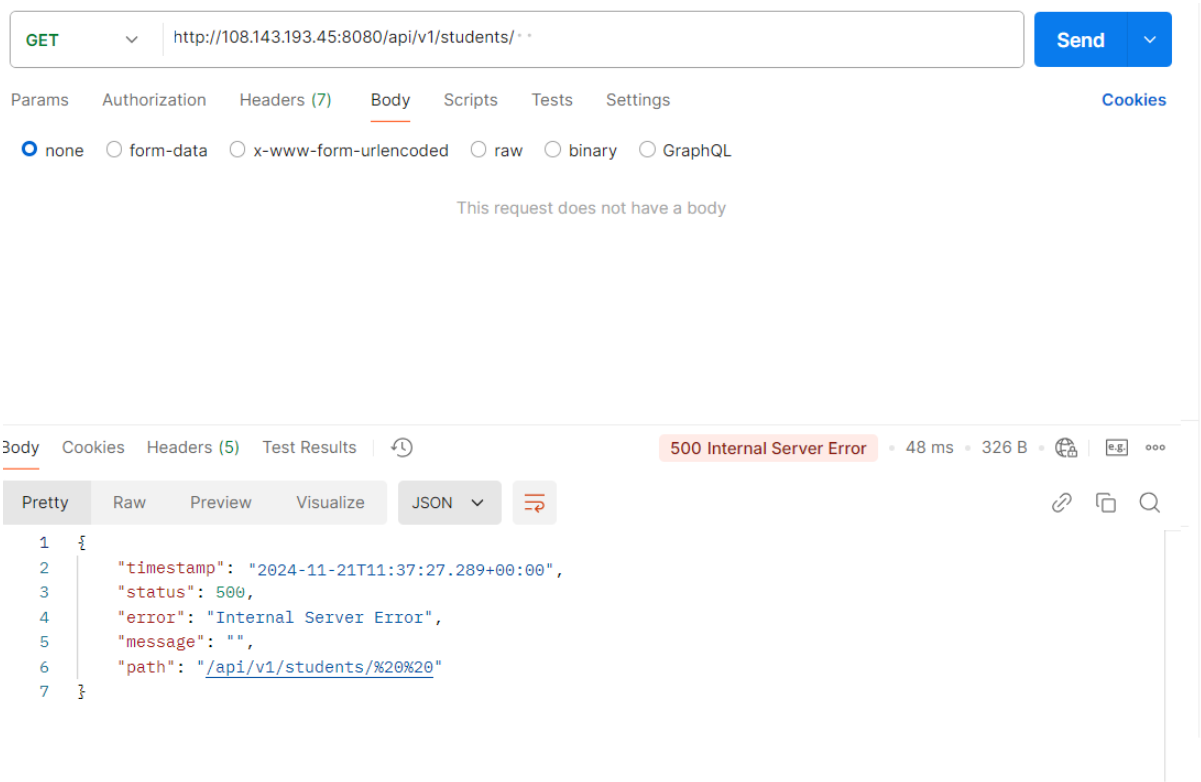
Název: Vrací se kód 500 při zadání mezer na místo id studenta
Závažnost: Major

Kroky k reprodukci chyby:

1. Do requestu GET zadám http://108.143.193.45:8080/api/v1/students/
2. Zmačknu tlačítko Send

Očekávaný výsledek:
Vrací se stavový kód 400.

Skutečný výsledek: Vrací se kód 500
Příloha:



Bug report 3

Název: Příjmení se v databázi ukládá velkými písmeny.
Závažnost: Minor
Kroky k reprodukci chyby:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data

"firstName":" Alexa",
"lastName": " Doležal",
"email": "a_doležal@gmail.com",
"age": 30

3. Zmačknu tlačítko Send

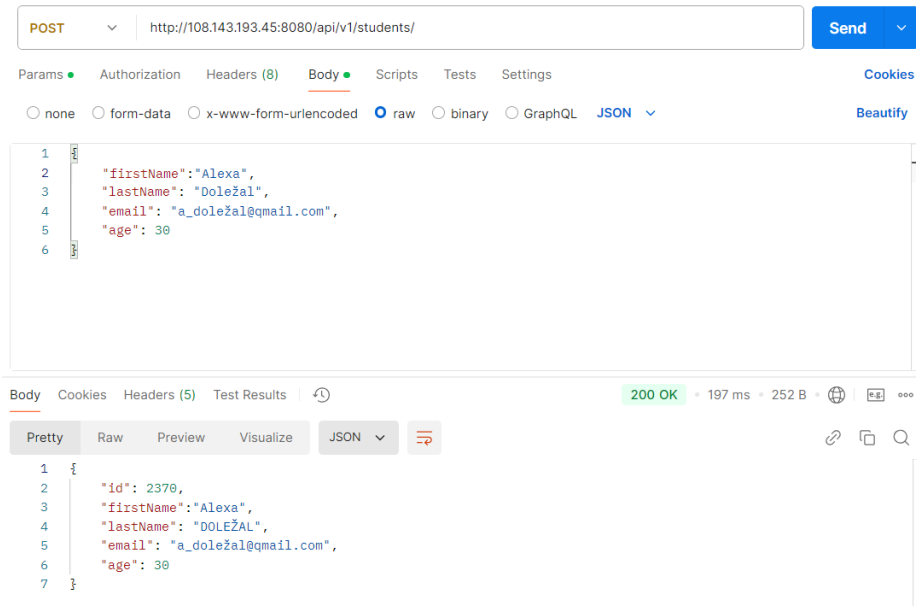
Očekávaný výsledek:
Vrací se stavový kód 200. Karta Body obsahuje JSON studenta, struktura kterého:

```
{
   "id":____,
   "firstName":" Alexa",
   "lastName": " Doležal",
   "email": "a_doležal@gmail.com",
   "age": 30
}
```

Skutečný výsledek: Vrací se stavový kód 200, ale příjmení se v databázi ukládá velkými písmeny

```
{
   "id":____,
   "firstName": " Alexa",
   "lastName": "DOLEŽAL",
   "email": " a_doležal@gmail.com",
   "age": 30
}
```

Příloha:



Bug report 4

Název: Vrací se kód 500 při vložení údajů do databáze, když není zadáno pouze firstName
Závažnost: Major

Kroky k reprodukci chyby:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data

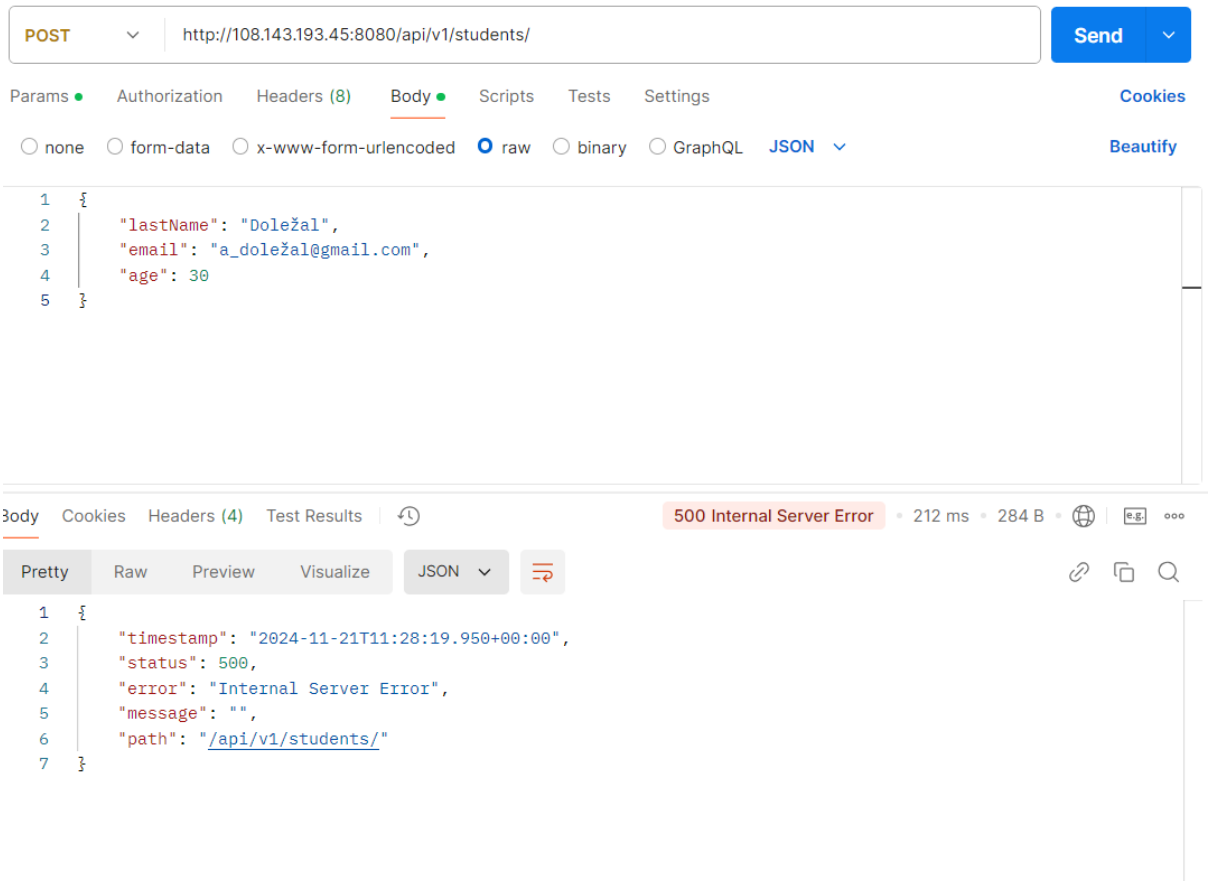
"lastName": "Doležal",
"email": "a_doležal@gmail.com",
"age": 30

3. Zmačknu tlačítko Send

Očekávaný výsledek:
Vrací se stavový kód 400.

Skutečný výsledek: Vrací se kód 500.

Příloha:



Bug report 5

Název: Vrací se kód 500 při vložení údajů do databáze, když není zadáno pouze lastName
Závažnost: Major

Kroky k reprodukci chyby:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data

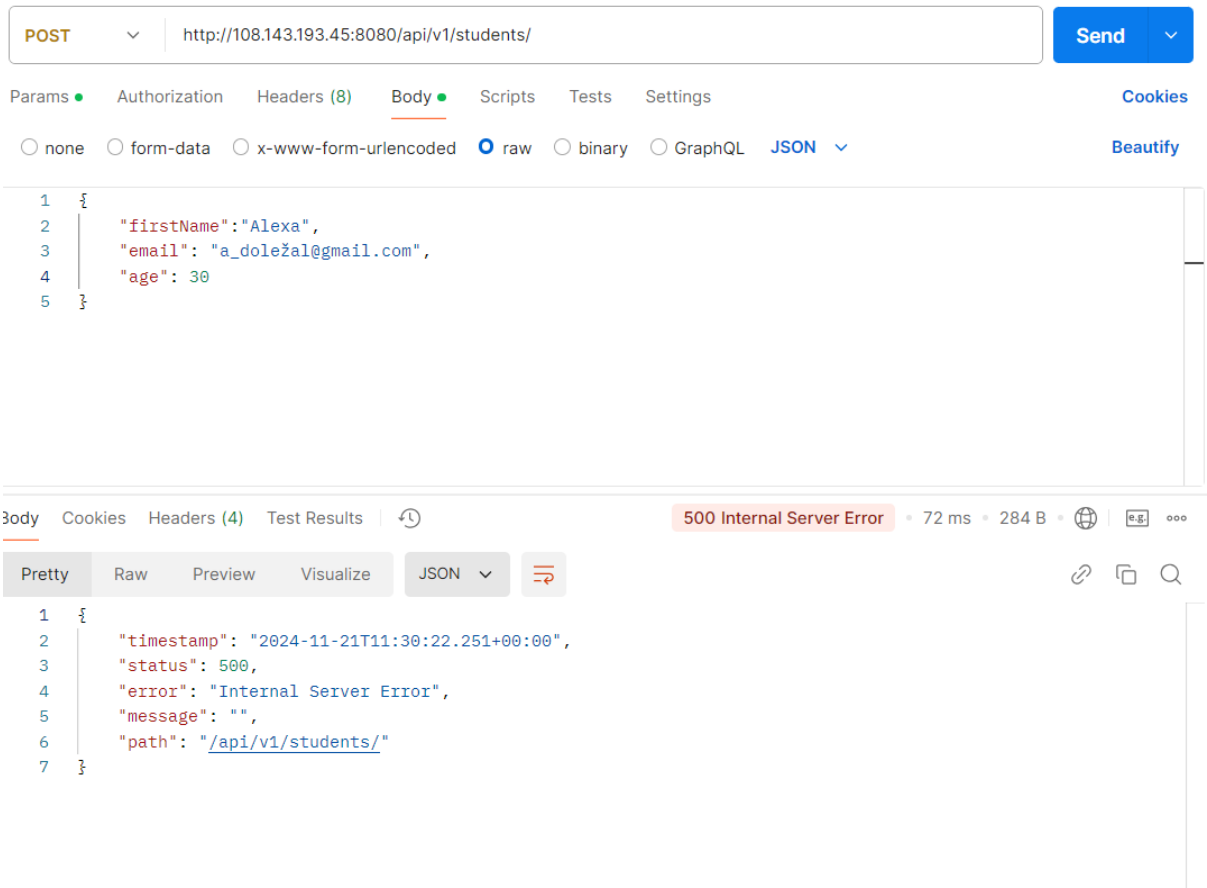
"firstName":"Alexa",
"email": "a_doležal@gmail.com",
"age": 30

3. Zmačknu tlačítko Send

Očekávaný výsledek:
Vrací se stavový kód 400.

Skutečný výsledek: Vrací se kód 500.

Příloha:



Bug report 6

Název: Vrací se kód 500 při vložení údajů do databáze, když není zadán pouze email
Závažnost: Major

Kroky k reprodukci chyby:

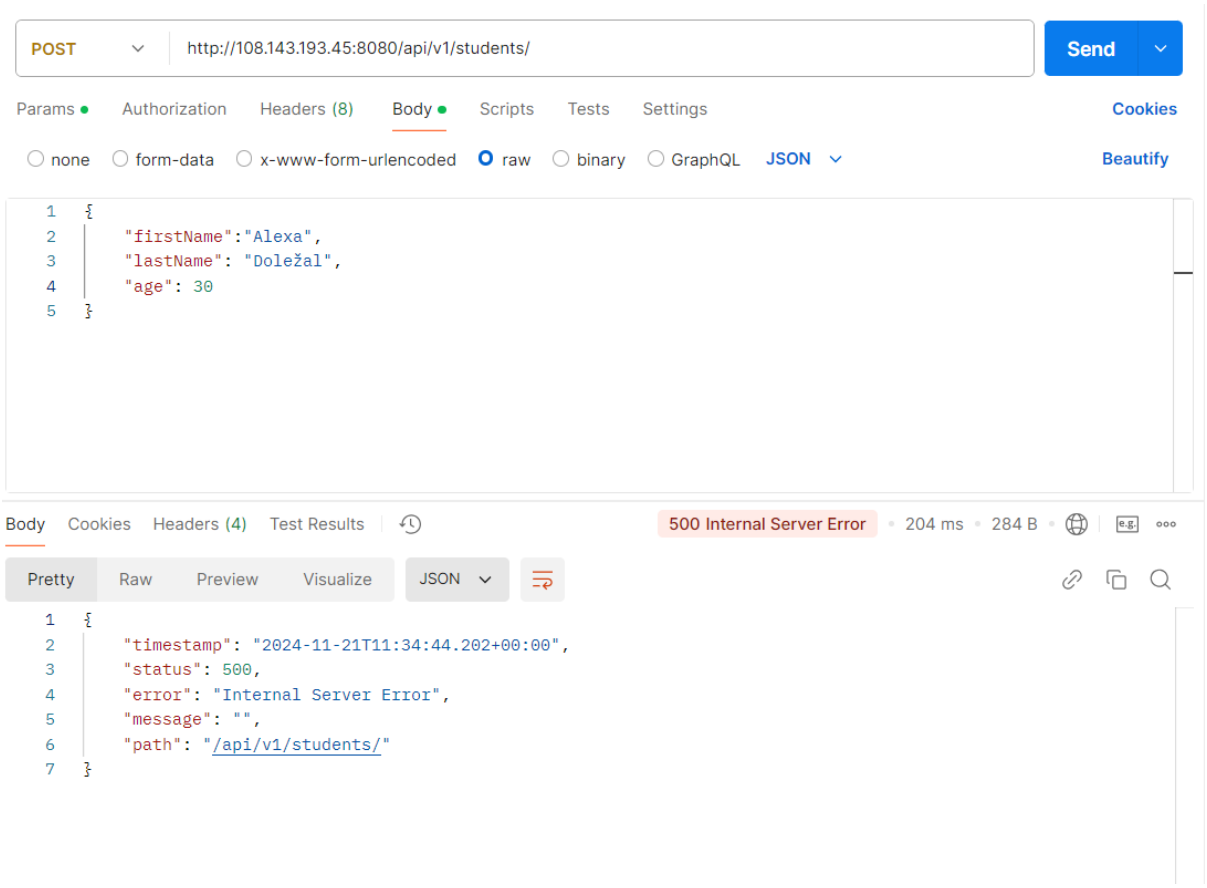
1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data

"firstName":"Alexa",
"lastName": "Doležal",
"age": 30

3. Zmačknu tlačítko Send

Očekávaný výsledek: Vrací se stavový kód 400.

Skutečný výsledek: Vrací se kód 500.
Příloha:



Bug report 7

Název: Vrací se kód 500 při vložení údajů do databáze, když není zadán věk
Závažnost: Major

Kroky k reprodukci chyby:

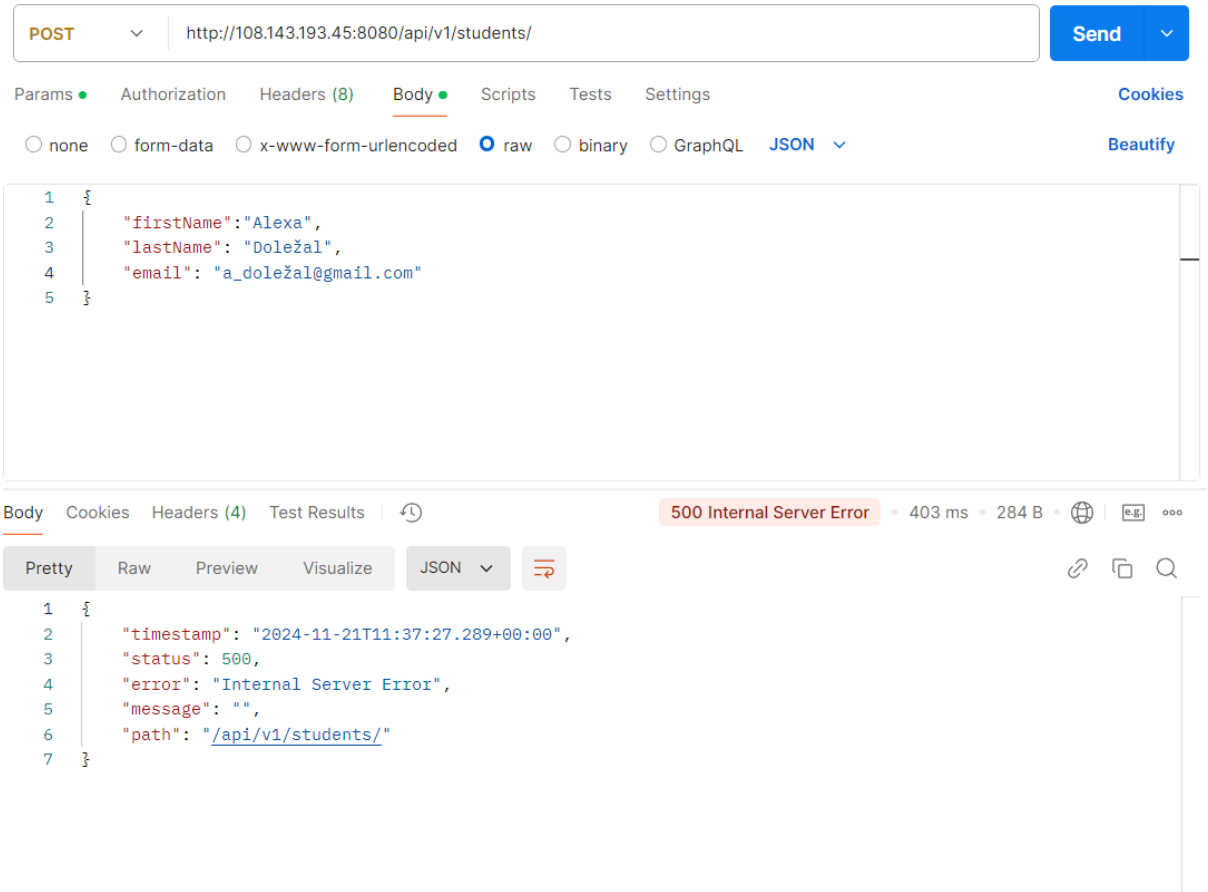
1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data

"firstName":"Alexa",
"lastName": "Doležal",
"email": "a_doležal@gmail.com"

3. Zmačknu tlačítko Send

Očekávaný výsledek: Vrací se stavový kód 400.

Skutečný výsledek: Vrací se kód 500.
Příloha:



Bug report 8

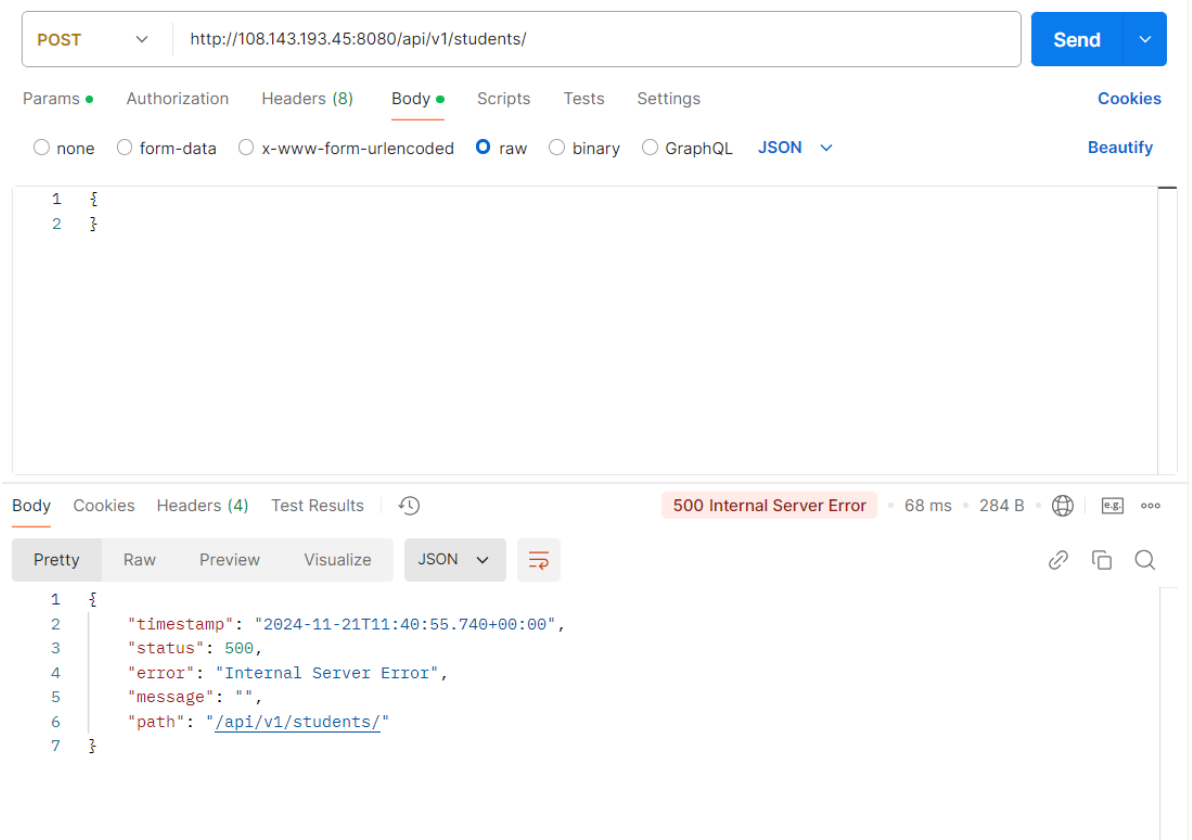
Název: Vrací se kód 500 při vložení údajů do databáze, když nejsou zadány žádné údaje
Závažnost: Major

Kroky k reprodukci chyby:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw ne zadám testovací data
3. Zmačknu tlačítko Send

Očekávaný výsledek: Vrací se stavový kód 400.

Skutečný výsledek: Vrací se kód 500.
Příloha:



Bug report 9

Název: Vložení údajů do databáze, když je zadáno speciální symboly do políčka firstName
Závažnost: Major

Kroky k reprodukci chyby:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data

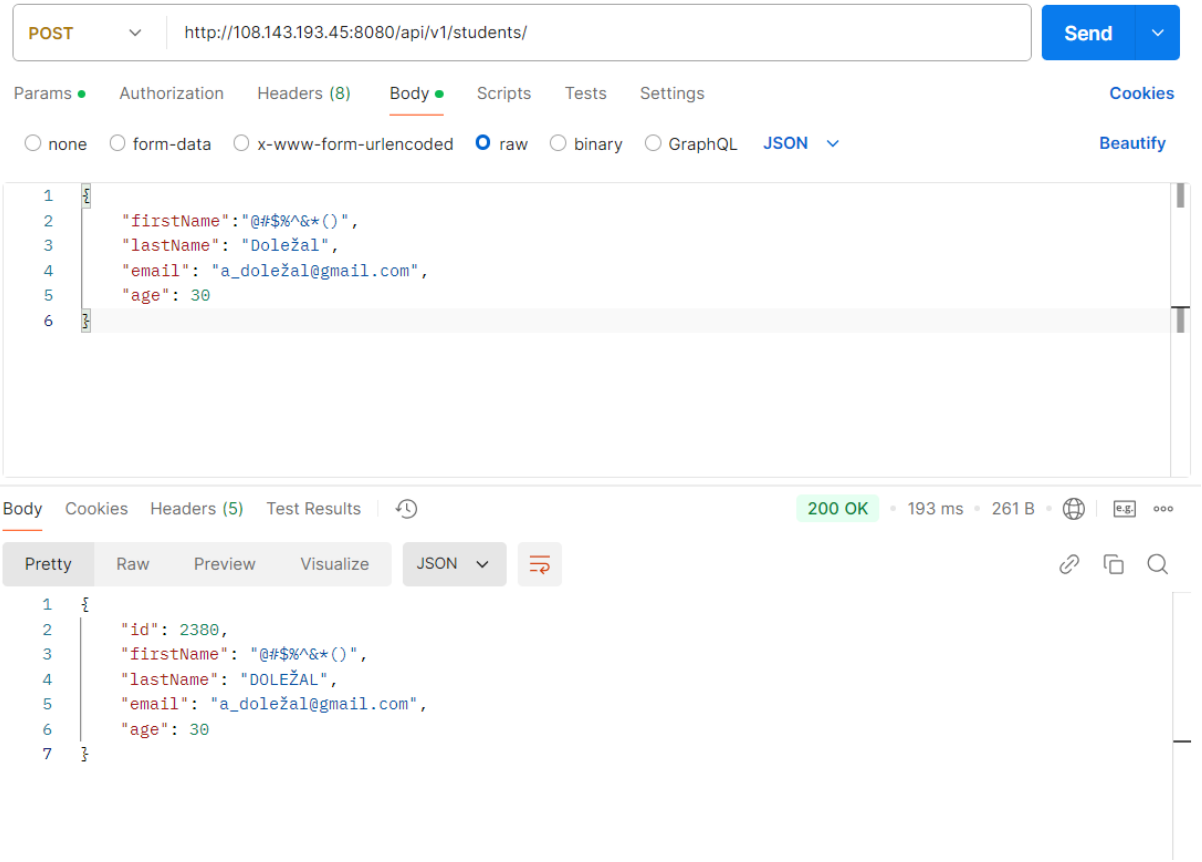
"firstName":"@#$%^&*()",
"lastName": "Doležal",
"email": "a_doležal@gmail.com",
"age": 30

3. Zmačknu tlačítko Send

Očekávaný výsledek:
Vrací se stavový kód 400.

Skutečný výsledek: Vrací se stavový kód 200.

Příloha:



Bug report 10

Název: Vložení údajů do databáze, když je zadáno mezery do políčka firstName

Závažnost: Major

Kroky k reprodukci chyby:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data

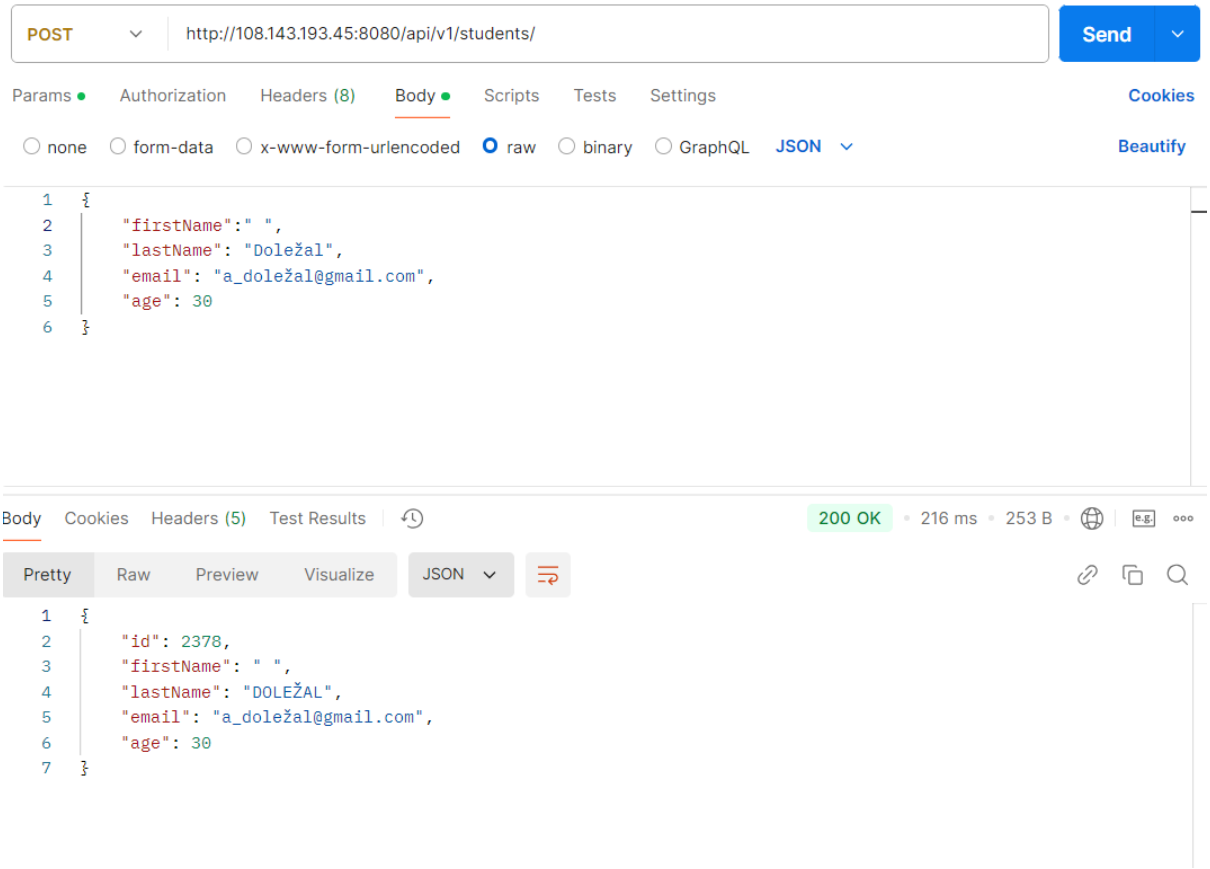
"firstName":" ",
"lastName": "Doležal",
"email": "a_doležal@gmail.com",
"age": 30

3. Zmačknu tlačítko Send

Očekávaný výsledek: Vrací se stavový kód 400.

Skutečný výsledek: Vrací se stavový kód 200.

Příloha:



Bug report 11

Název: Vložení údajů do databáze, když je zadáno číslice do políčka firstName
Závažnost: Major

Kroky k reprodukci chyby:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data

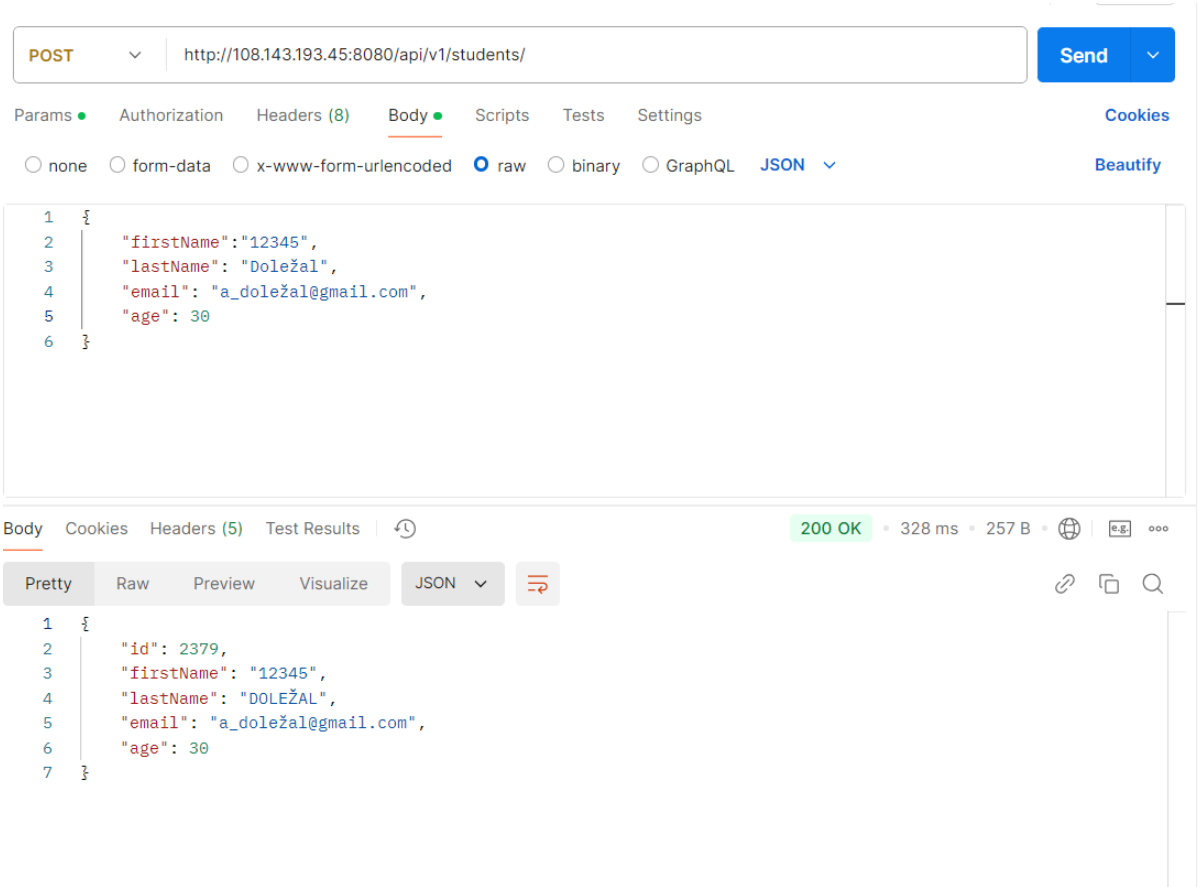
"firstName":"12345",
"lastName": "Doležal",
"email": "a_doležal@gmail.com",
"age": 30

3. Zmačknu tlačítko Send

Očekávaný výsledek:
Vrací se stavový kód 400.

Skutečný výsledek: Vrací se stavový kód 200.

Příloha:



Bug report 12

Název: Vložení údajů do databáze, když je zadáno údaj délkou 256 symbolu do políčka firstName.
Závažnost: Minor

Kroky k reprodukci chyby:

1. Do requestu POST zadám http://108.143.193.45:8080/api/v1/students/
2. Do políčka raw zadám testovací data

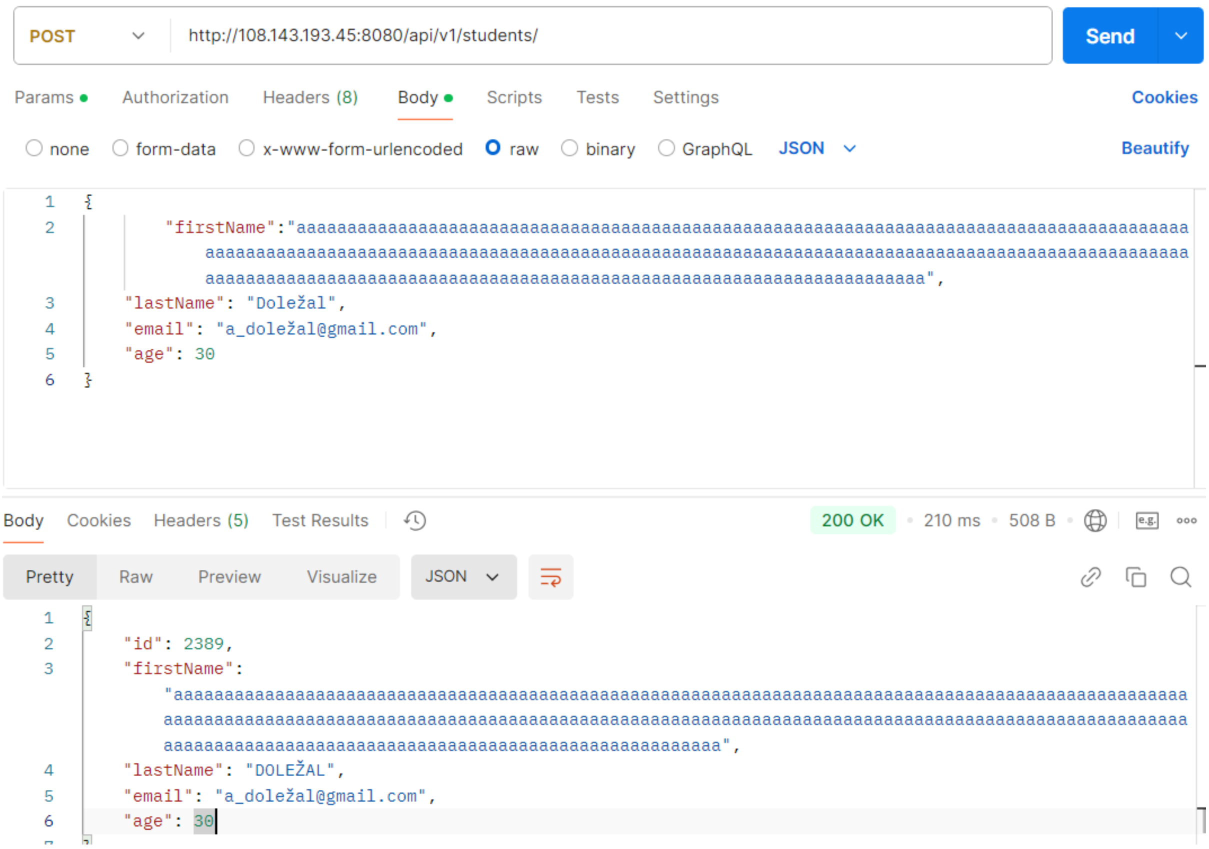
"firstName":"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa",
"lastName": "Doležal",
"email": "a_doležal@gmail.com",
"age": 30

3. Zmačknu tlačítko Send

Očekávaný výsledek: Vrací se stavový kód 400.

Skutečný výsledek: Vrací se stavový kód 200.

Příloha:



Bug report 13

Název: Vrací se kód 500 při pokusu smazat data neplatného id studenta
Závažnost: Major

Kroky k reprodukci chyby:

1. Do requestu DELETE zadám http://108.143.193.45:8080/api/v1/students/888
2. Zmačknu tlačítko Send

Očekávaný výsledek: Vrací se stavový kód 404.

Skutečný výsledek: Vrací se kód 500.

Příloha: