FINÁLNÍ PROJEKT č.1



Autor: Ing. Zuzana Štrbová Datum: 18. června 2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	7
BUG REPORT	11

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
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.

1. **Abstrakt:** Získání dat o existujícícím studentovi

Příprava dat: Vybrat existujícího studenta s ID = 403

```
SELECT * FROM student WHERE id= 403;
```

("403","40","hboss@testmail.com","Hugo","BOSS")

Kroky: 1) GET http://108.143.193.45:8080/api/v1/students/403

2) Potvrdit odeslání - Send

Očekávaný výsledek: 1) Status code 200 OK

2) výsledek:

```
"id": 403,
"firstName": "Hugo",
"lastName": "BOSS",
"email": "hboss@testmail.com",
"age": 40
```

2. **Abstrakt:** Získání dat o neexistujícím studentovi

Příprava dat: Vybrat neexistujícího studenta s ID = X

SELECT * FROM student WHERE id = X;
(Error Code: 1054, Unknown column "X")

Kroky: 1) GET http://108.143.193.45:8080/api/v1/students/X

2) Potvrdit odeslání - Send

Očekávaný výsledek: 1) Status code 400 Bad Request

2) Výsledek:

Žádná data nenalezena, nekorektně zadaný požadavek.

3. **Abstrakt:** Získání dat o neexistujícím studentovi

Příprava dat: Vybrat neexistujícího studenta s ID = 0

SELECT * FROM student WHERE id = 0;

(0 row(s) returned)

Kroky: 1) GET http://108.143.193.45:8080/api/v1/students/0

2) Potvrdit odeslání - Send

Očekávaný výsledek: 1) Status code 404 Not Found

2) Výsledek:

Žádná data nenalezena, neexistující student.

4. Abstrakt: Smazání dat o existujícím studentovi Příprava dat: Vybrat existujícího studenta s ID = 338 SELECT * FROM student WHERE id = 338; ("338","21","pepaomacka@gmail.com","Pe","OMACKA") Kroky: 1) DELETE http://108.143.193.45:8080/api/v1/students/338 2) Potvrdit odeslání - Send Očekávaný výsledek: 1) Status code 200 OK 2) Výsledek: Data daného studenta smazána. 3) Data úspěšně smazána z databáze – uvedený záznam s id = 338 neexistuie. 5. Abstrakt: Smazání dat o neexistujícím studentovi Příprava dat: Vybrat neexistujícího studenta s ID = 20000 SELECT * FROM student WHERE id = 20000; (0 row(s) returned) Kroky: 1) DELETE http://108.143.193.45:8080/api/v1/students/20000 2) Potvrdit odeslání - Send Očekávaný výsledek: 1) Status code 404 Not Found 2) Výsledek: Výmaz nebylo možné provést, neexistující záznam. Úspěšné založení nového studenta 6. Abstrakt: Kroky: 1) Zadání body JSON "firstName": "James", "lastName": "Bond", "email": "jb008@email.com", "age": 20 2) POST http://108.143.193.45:8080/api/v1/students/ 3) Potvrdit odeslání - Send

Očekávaný výsledek:

- 1) Status code 200 OK
- 2) Výsledek:

Založení studenta, vygenerování dat

```
"id": ,
              --automaticky dogenerováno konkrétní id ("číslo")
"firstName": "james",
"lastName": "bond",
"email": "jb008@email.com",
"age": 20
```

- 3) Data úspěšně vložena do databáze uvedený záznam s přiřazeným id existuje.
- 4) Data jsou v databázi uložena ve správném formátu (lowerCase).

7. **Abstrakt**: Neúspěšné založení nového studenta s chybným emailem

Kroky: 1) Zadání body JSON s nekorektně zadaným emailem

{
"firstName": "James",
"lastName": "Bond",
"email": "email.com",
"age": 20
}

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

3) Potvrdit odeslání - Send

Očekávaný výsledek:

- 1) Status code 400 Bad Request, email must be in format *@*.*
- 2) Výsledek:

Student nebude založen.

8. Abstrakt:

Neúspěšné založení nového studenta s chybným formátem věku

Kroky:

1) Zadání body JSON s nekorektně zadaným formátem věku/věk obsahuje písmena

```
{
"firstName": "firstName",
"lastName": "lastName",
"email": "email@email.com",
"age": "age"
}
```

- 2) POST http://108.143.193.45:8080/api/v1/students/
- 3) Potvrdit odeslání Send

Očekávaný výsledek:

- 1) Status code 400 Bad Request, age must be in numeric format
- 2) Výsledek:

Student nebude založen. (Age je povinný údaj, nelze uložit chybný věk ani jej ponechat nevyplněný.)

9. Abstrakt:

Použití metody POST s HTTP odkazujícím na konkrétního klienta k založení nového klienta

Příprava dat:

```
SELECT * FROM student WHERE id = 296;
("296","0","karolko@gmail.com","KAROL","UNTERWASSERMAN)
```

Kroky:

1) Zadání body JSON s korektním vyplněním všech povinných položek

```
{
"firstName": "firstName",
"lastName": "lastName",
"email": "email@email.com",
"age": 20
```

- 2) POST http://108.143.193.45:8080/api/v1/students/296
- 3) Potvrdit odeslání Send

Očekávaný výsledek:

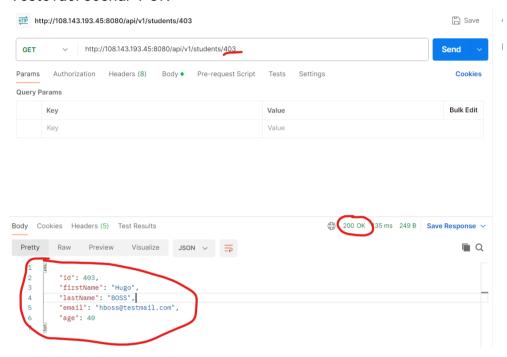
- 1) Status code 405 Method Not Allowed
- 2) Výsledek:

Student nebude založen.

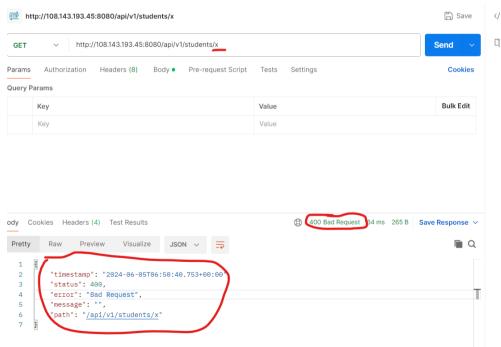
EXEKUCE TESTŮ

Testovací scénáře jsem provedla, přikládám výsledky testů.

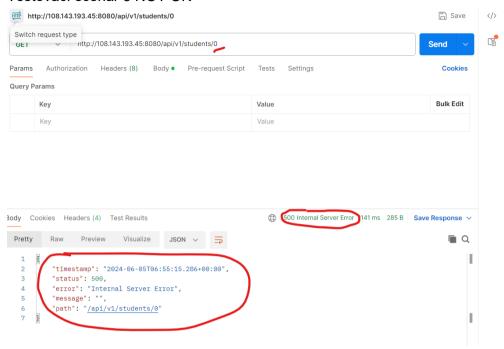
Testovací scénář 1 OK



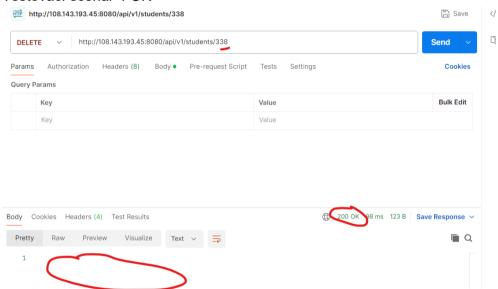
Testovací scénář 2 OK



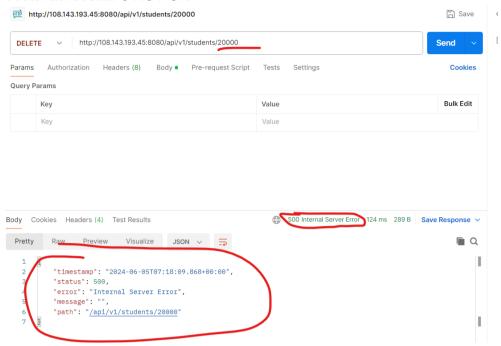
Testovací scénář 3 NOT OK



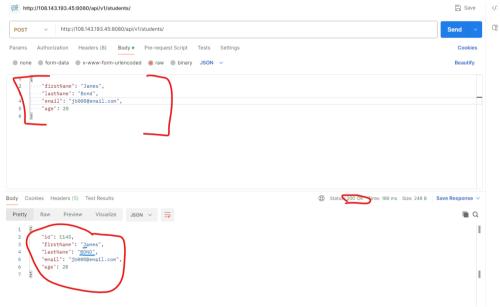
Testovací scénář 4 OK



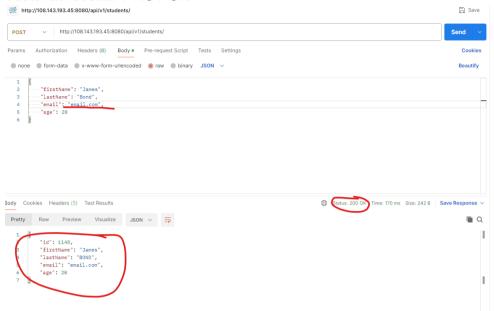
Testovací scénář 5 NOT OK



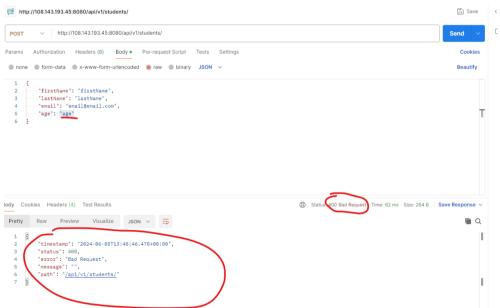
Testovací scénář 6 NOT OK



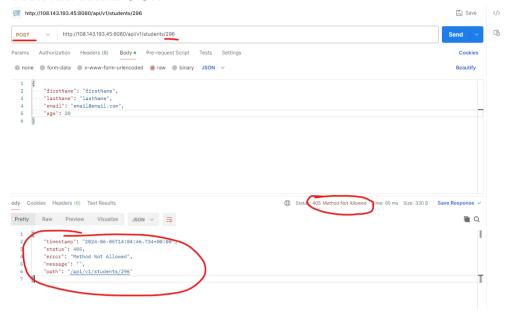
Testovací scénář 7 NOT OK



Testovací scénář 8 OK



Testovací scénář 9 OK



BUG REPORT

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

Testovací scénář 3:

špatný kód chyby
 (skutečný výsledek – status code 500 Internal Server Error)

Testovací scénář 5:

 špatný kód chyby (skutečný výsledek – status code 500 Internal Server Error)

Testovací scénář 6:

- data firstName a lastName nejsou uložena v požadovaném formátu lowerCase
- vložení studenta do databáze s odlišným formátem (od požadovaného) jména a příjmení (skutečný výsledek - data:

```
"id":1145,
"firstName": "James",
"lastName": "BOND",
"email": "jb008@email.com",
"age": 20
```

}

Testovací scénář 7:

- špatný kód (skutečný výsledek - status code 200 OK)
- založení studenta s nekorektním formátem emailu
- uložený studenta s chybným emailem do databáze (skutečný výsledek - data:

```
{
"id": 1148,
"firstName": "James",
"lastName": "BOND",
"email": "email.com",
"age": 20
} )
```