

FINÁLNÍ PROJEKT

č.1



Autor: Filip Rozum (Discord: F.Rozum/Denizer)
Datum: 24.6.2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	5
BUG REPORT	5

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: sgqcu2878p6q2zzw5tw4 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.

1.GET

*příprava dat: ID existujícího studenta = 961, SELECT * FROM student WHERE id=961*

abstrakt: Ukaž údaje existujícího studenta

Postup: Zadej adresu http://108.143.193.45:8080/api/v1/students/961, vyber možnost GET a klikni na Send

Očekávaný výsledek:- status code 200 OK

- odpověď: { "id": 961,
 "firstName" : "John",
 "lastName": "DOE",
 "email": "JohnDoe@speed.org",
 "age": 19
 }

Test pro neexistujícího studenta: Opakuj postup s id neexistujícího studenta

Očekávaný výsledek - status code 404

2.DELETE

*příprava dat: ID existujícího studenta = 755, SELECT * FROM student WHERE id=755*

abstrakt: Vymaž všechny údaje existujícího studenta

Postup: Zadej adresu http://108.143.193.45:8080/api/v1/students/755, vyber možnost DELETE a klikni na Send

Očekávaný výsledek- status code 200 OK

- All data about particular student are REMOVED

-Poté kontrola smazání údajů v databázi

Test pro neexistujícího studenta: Opakuj postup s id neexistujícího studenta

Očekávaný výsledek - status code 404

3.POST

příprava dat: Vymyslet data pro nového studenta (firstName = Vernon, lastName = Roach, email = vroach@stripes.com, age = 34)

abstrakt: Přidej nového studenta

Postup: -Zadej adresu <http://108.143.193.45:8080/api/v1/students/> a zvol možnost POST.

-Zadej request (Body, raw, JSON) - {

```
"firstName": "Vernon",  
"lastName": "Roach",  
"email": "vroach@stripes.com",  
"age": 34
```

```
}
```

-Klikni na Send

Očekávaný výsledek- status code 201 created

- odpověď: {

```
"id": ***,  
"firstName": "Vernon",  
"lastName": "Roach",  
"email": "vroach@stripes.com",  
"age": 34
```

```
}
```

-Poté kontrola přidaného studenta v databázi

-Pro úspěšné založení studenta je nutné zadat všechny atributy (jinak status code 400 bad request)

-age musí být 1-150 let, firstName a lastName 3-50 znaků, email ve formátu @.***
- jinak status code 400 bad request

-Všechny atributy se do databáze mají ukládat formou lowerCase

EXEKUCE TESTŮ

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

GET - existující student

GET

http://108.143.193.45:8080/api/v1/students/961

Params

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

200 OK

Pretty

Raw

Preview

Visualize

```
1 {
2   "id": 961,
3   "firstName": "John",
4   "lastName": "DOE",
5   "email": "JohnDoe@speed.org",
6   "age": 19
7 }
```

GET - neexistující student

GET

http://108.143.193.45:8080/api/v1/students/235

Params

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

500 Internal ... 135 ms 287 B Save

Pretty

JSON

```
1 {
2   "timestamp": "2024-06-21T14:49:1
3     +00:00",
4   "status": 500,
5   "error": "Internal Server Error"
6   "message": "",
7   "path": "/api/v1/students/235"
8 }
```

SELECT * FROM student WHERE id=235;

2

Result Grid

Filter Rows:

Export:

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

DELETE - existující student

DELETE ▼ | http://108.143.193.45:8080/api/v1/students/755

Params ▼ ... | Body ▼ 🌐 200 OK 1

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Pretty ▼ | Text ▼ | 🔄

1

1 • SELECT * FROM student WHERE id=755;
2

Result Grid | 📊 | ↕ Filter Rows: | Export

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

DELETE - neexistující student

1 • SELECT * FROM student WHERE id=405;
2

Result Grid | 📊 | ↕ Filter Rows: | Export

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

DELETE ▼ | http://108.143.193.45:8080/api/v1/students/405 Send

Params ▼ ... | Body ▼ 🌐 500 Internal Se... 105 ms 287 B Save Resp...

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Pretty ▼ | JSON ▼ | 🔄

```
1
2   "timestamp": "2024-06-24T09:00:46.862
3     +00:00",
4   "status": 500,
5   "error": "Internal Server Error",
6   "message": "",
7   "path": "/api/v1/students/405"
```

POST

POST ▼ http://108.143.193.45:8080/api/v1/students/ ▶

Body ▼ ... Body ▼ 200 OK 160 ms 253 B Save

raw ▼ JSON ▼ Beautify Pretty ▼ JSON ▼ ↺

```
1 {
2   "firstName": "Vernon",
3   "lastName": "Roach",
4   "email": "vroach@stripes.com",
5   "age": 34
6 }
```

```
1 {
2   "id": 1276,
3   "firstName": "Vernon",
4   "lastName": "ROACH",
5   "email": "vroach@stripes.com",
6   "age": 34
7 }
```

Limit to 1000 rows

```
1 • SELECT * FROM student WHERE first_name="Vernon";
2
```

Result Grid 📄 Filter Rows: Export: 📄 Wrap Cell

	id	age	email	first_name	last_name
▶	1276	34	vroach@stripes.com	Vernon	ROACH

POST - další testy

1. chybí atribut

POST ▼ http://108.143.193.45:8080/api/v1/students/ ▶

Body ▼ ... Body ▼ 500 Internal Se... 180 ms 284 B Save R

raw ▼ JSON ▼ Beautify Pretty ▼ JSON ▼ ↺

```
1 {
2   "firstName": "RENE",
3   "lastName": "Reno",
4   "age": 35
5 }
```

```
1 {
2   "timestamp": "2024-06-24T09:37:57.1+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "message": "",
6   "path": "/api/v1/students/"
7 }
```


2. nepovolený počet znaků ve firstName


Body ▼ 200 OK 17

Pretty Raw Preview Visualize JSON ▼ ↺

```
1 {
2   "id": 1279,
3   "firstName": "Li",
4   "lastName": "RENO",
5   "email": "ahfnf@hbg.cs",
6   "age": 35
7 }
```


3. nepovolený věk

Body  200 OK 1

Pretty Raw Preview Visualize JSON 

```
1 {
2   "id": 1280,
3   "firstName": "Dante",
4   "lastName": "RENO",
5   "email": "NeTRo@hbg.cs",
6   "age": 235
7 }
```

4. nepovolený počet znaků v lastName

Body Cookies Headers (5) Test Results  200 OK 180

Pretty Raw Preview Visualize JSON 

```
1 {
2   "id": 1281,
3   "firstName": "Dante",
4   "lastName": "QWERTZUIOPQWERTZUIOPQWERTZUIOPQWERTZUIOPQWERT",
5   "email": "oprrr@hbg.cs",
6   "age": 18
7 }
```

5. nepovolený formát emailu



Body Cookies Headers (5) Test Results  200 OK

Pretty Raw Preview Visualize JSON 

```
1 {
2   "id": 1282,
3   "firstName": "Hugo",
4   "lastName": "RENO",
5   "email": "landor@",
6   "age": 47
7 }
```

6. kontrola ukládání dat do databáze

```
1 • SELECT * FROM student WHERE last_name="Reno";
2
```

Result Grid  Filter Rows: Export:  Wrap C

	id	age	email	first_name	last_name
▶	1279	35	ahfnf@hbg.cs	Li	RENO
	1280	235	NeTRo@hbg.cs	Dante	RENO

BUG REPORT

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

GET - Při zobrazení neexistujícího studenta - status code 500 (má být 404)

DELETE - Při smazání studenta chybí odezva "All data about particular student are REMOVED"

- Při mazání neexistujícího studenta - status code 500 (má být 404)

POST a ukládání dat do databáze

- Při založení studenta - status code 200 OK (má být 201 Created)
- Při chybějícím atributu - status code 500 (má být 400)
- U všech případů špatně zadaného atributu byl student úspěšně založen (status code 200 OK) - Má se zobrazit status code 400 bad request
- Databáze neukládá data v lowerCase
 - email a firstName zůstává v původní formě
 - lastName se ukládá v upperCase