

FINÁLNÍ PROJEKT

č.1



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

Okomentoval(a): [ŠZ1]: zuzka2414

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.

- Abstrakt:** Získání dat o existujícím 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`
- 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.
- 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 neexistuje.
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.
6. **Abstrakt:** Úspěšné založení nového studenta
- 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

http://108.143.193.45:8080/api/v1/students/403

GET http://108.143.193.45:8080/api/v1/students/403 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (5) Test Results 200 OK 35 ms 249 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 403,
3   "firstName": "Hugo",
4   "lastName": "BOSS",
5   "email": "hboss@testmail.com",
6   "age": 40
7 }
```

Testovací scénář 2 OK

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

GET http://108.143.193.45:8080/api/v1/students/x Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (4) Test Results 400 Bad Request 54 ms 265 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2024-06-05T06:50:49.753+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "",
6   "path": "/api/v1/students/x"
7 }
```

Testovací scénář 3 NOT OK

http://108.143.193.45:8080/api/v1/students/0

Switch request type **GET** http://108.143.193.45:8080/api/v1/students/0 **Send**

Params Authorization Headers (8) Body **Pre-request Script** Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (4) Test Results **500 Internal Server Error** 141 ms 285 B **Save Response**

Pretty Raw Preview Visualize JSON **Copy**

```
1 {
2   "timestamp": "2024-06-05T06:55:15.286+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "message": "",
6   "path": "/api/v1/students/0"
7 }
```

Testovací scénář 4 OK

http://108.143.193.45:8080/api/v1/students/338

DELETE http://108.143.193.45:8080/api/v1/students/338 **Send**

Params Authorization Headers (8) Body **Pre-request Script** Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (4) Test Results **200 OK** 68 ms 123 B **Save Response**

Pretty Raw Preview Visualize **Text** **Copy**

```
1
```


Testovací scénář 5 NOT OK

http://108.143.193.45:8080/api/v1/students/20000

DELETE http://108.143.193.45:8080/api/v1/students/20000 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (4) Test Results 500 Internal Server Error 124 ms 289 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2024-06-05T07:18:09.868+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "message": "",
6   "path": "/api/v1/students/20000"
7 }
```

Testovací scénář 6 NOT OK

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

POST http://108.143.193.45:8080/api/v1/students/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

@ none @ form-data @ x-www-form-urlencoded raw @ binary JSON Beautify

```
1 {
2   "firstName": "James",
3   "lastName": "Bond",
4   "email": "j0000@email.com",
5   "age": 20
6 }
```

Body Cookies Headers (5) Test Results Status: 500 106 ms Size: 248 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1540,
3   "firstName": "James",
4   "lastName": "Bond",
5   "email": "j0000@email.com",
6   "age": 20
7 }
```

Testovací scénář 7 NOT OK

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

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "firstName": "James",
3   "lastName": "Bond",
4   "email": "email.com",
5   "age": 28
6 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 170 ms Size: 242 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "1dab",
3   "firstName": "James",
4   "lastName": "BOND",
5   "email": "email.com",
6   "age": 28
7 }
```

Testovací scénář 8 OK

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

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

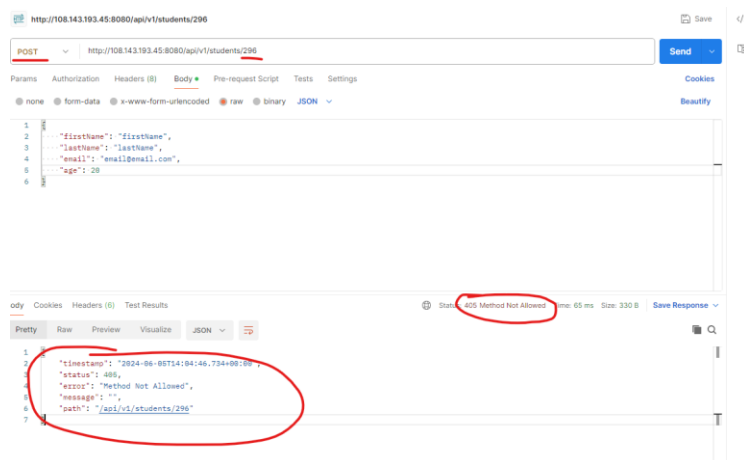
```
1 {
2   "firstName": "firstName",
3   "lastName": "lastName",
4   "email": "email@email.com",
5   "age": "age"
6 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 62 ms Size: 264 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2024-06-05T13:45:46.470+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "",
6   "path": "/api/v1/students/"
7 }
```

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
- uložení studenta s chybným emailem do databáze
(skutečný výsledek - data:

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