

FINÁLNÍ PROJEKT

č.1



Autor: Kateřina Jetelinová
Datum: 14.8.2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE.....	4
EXEKUCE TESTŮ.....	18
BUG REPORT.....	34

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: *** Host: *** Username: *** Password: ***
REST-API	***

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.
Všechny scénáře jsou vykonávány v nástroji Postman, MySQLWorkbench.

1, Metoda GET

ID test case: tcget.01

Název: Získání dat o existujícím studentovi s ID 712

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- firstName a lastName mohou obsahovat 3-50 znaků
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu GET	Objeví se zelený nápis GET
2, Vedle GET do volného pole vložíme odkaz: http://108.143.193.45:8080/api/v1/students/712	Vložený odkaz se propíše do daného pole
3, Stiskneme tlačítko SEND	Objeví se hláška „Status code 200 OK“
4, Zkontrolujeme údaje	V dolní části se objeví údaje: { "id": 712, "firstName": "Armando", "lastName": "Biely", "email": armandbi@hismail.com , "age": 120 }

2, Metoda GET

ID test case: tcget.02

Název: (Ne)získání dat o studentovi s ID 50, který v databázi neexistuje

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- firstName a lastName mohou obsahovat 3-50 znaků
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu GET	Objeví se zelený nápis GET
2, Vedle GET vložíme do volného pole odkaz: http://108.143.193.45:8080/api/v1/students/50	Vložený odkaz se propíše do daného pole
3, Stiskneme tlačítko SEND	Objeví se hláška „404 Not found“

3, Metoda POST

ID test case: tcpost.01

Název: Vytvoření nového studenta se všemi správnými atributy (hraniční věk 1 rok)

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- firstName a lastName mohou obsahovat 3-50 znaků
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu POST	Objeví se nám žlutý nápis POST
2, Vložíme do vedlejšího volného pole tento odkaz: http://108.143.193.45:8080/api/v1/students/	Odkaz se propíše do řádku
3, O řádek níže zvolíme "Body"	Proklikneme se, pod se nám ukáže další nabídka a okénko ke psaní
4, Z nabídky si vybereme "raw" a formát "JSON"	Navolíme požadované formáty
5, Do okna vložíme následující data: { "firstName": "Bobby", "lastName": "Stanley", "email": "bs007@yamail.com", "age": 1 }	Do okna se nám propíší požadovaná data
6, Stiskneme tlačítko Send	Objeví se status kód "201 Created", studentovi bude přiděleno ID

4, Metoda POST

ID test case: tcpost.02

Název: Pokus o vytvoření nového studenta s lastName a firstName v nesprávném formátu (2 znaky)

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- **firstName a lastName mohou obsahovat 3-50 znaků**
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu POST	Objeví se nám žlutý nápis POST
2, Vložíme do vedlejšího volného pole tento odkaz: http://108.143.193.45:8080/api/v1/students/	Odkaz se propíše do řádku
3, O řádek níže zvolíme „Body“	Proklikneme se, pod se nám objeví další nabídka a okénko ke psaní
4, Z nabídky si vybereme „raw“ a formát „JSON“	Navolíme požadované formáty
5, Do okna vložíme data: { “firstName”: “Yi”, “lastName”: “Yu”, “email”: “iu@email.com”, “age”: 29 }	Data se propíší
6, Stiskneme tlačítko Send	Objeví se status kód „400 Bad request“

5, Metoda POST

ID test case: tcpost.03

Název: Pokus o vytvoření studenta s firstName a lastName ve správném formátu (hraniční hodnota – 3 znaky)

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- **firstName a lastName mohou obsahovat 3-50 znaků**
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu POST	Objeví se nám žlutý nápis POST
2, Vložíme do vedlejšího volného pole tento odkaz: http://108.143.193.45:8080/api/v1/students/	Odkaz se propíše do řádku
3, O řádek níže zvolíme „Body“	Proklikneme se, pod se nám objeví další nabídka a okénko ke psaní
4, Z nabídky si vybereme „raw“ a formát „JSON“	Navolíme požadované formáty
5, Do okna vložíme data: { "firstName": "Yin", "lastName": "Yun", "email": "yy3@mailito.com", "age": 31 } 6, Stiskneme tlačítko Send	Data se propíší Objeví se status kód „201 Created“

6, Metoda POST

ID test case: tcpost.04

Název: Pokus o vytvoření nového studenta s firstName a lastName v nesprávném formátu (přes 50 znaků)

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- **firstName a lastName mohou obsahovat 3-50 znaků**
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu POST	Objeví se nám žlutý nápis POST
2, Vložíme do vedlejšího volného pole tento odkaz: http://108.143.193.45:8080/api/v1/students/	Odkaz se propíše do řádku
3, O řádek níže zvolíme „Body“	Proklikneme se, pod se nám objeví další nabídka a okénko ke psaní
4, Z nabídky si vybereme „raw“ a formát „JSON“	Navolíme požadované formáty
5, Do okna vložíme dle výběru vždy jednoho studenta: { “firstName”: “Chargogg- agoggmanchauggagoggchaubuna- gungamauggaaaaaa“, “lastName”: “Llanfairpwllgwyngyllgogery- chwyrndrobwllllantysiliogogogoch”, “email”: “lakecity@yahoo.com”, “age”: 120 }	Data se propíší
6, Stiskneme tlačítko Send	Objeví se status kód „400 Bad request“

7, Metoda POST

ID test case: tcpost.05

Název: Pokus o vytvoření nového studenta s firstName a lastName ve správném formátu (hraniční hodnota – 50 znaků)

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- **firstName a lastName mohou obsahovat 3-50 znaků**
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu POST	Objeví se nám žlutý nápis POST
2, Vložíme do vedlejšího volného pole tento odkaz: http://108.143.193.45:8080/api/v1/students/	Odkaz se propíše do řádku
3, O řádek níže zvolíme „Body“	Proklikneme se, pod se nám objeví další nabídka a okénko ke psaní
4, Z nabídky si vybereme „raw“ a formát „JSON“	Navolíme požadované formáty
5, Do okna vložíme data: { “firstName”: “Bernadetteevangelinejacquelinemarylouisewilhelmina” “lastName”: “Alessandroflorentinomontgomeryrutherfordwashington”, “email”: “ba50@email.com”, “age”: 60 }	Data se propíší
6, Stiskneme tlačítko Send	Objeví se status kód „201 Created“

8, Metoda POST

ID test case: tcpost.06

Název: Pokus o založení nového studenta s nesprávným věkem (151 let)

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- **Rozpětí pro věk je 1-150 let**
- firstName a lastName mohou obsahovat 3-50 znaků
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Otevřeme Postmana	Program se otevře
2, Zvolíme metodu POST	Objeví se nám žlutý nápis POST
3, Vložíme do vedlejšího volného pole tento odkaz: http://108.143.193.45:8080/api/v1/students/	Odkaz se propíše do řádku
4, O řádek níže zvolíme „Body“	Proklikneme se, pod se nám objeví další nabídka a okénko ke psaní
5, Z nabídky si vybereme „raw“ a formát „JSON“	Navolíme požadované formáty
6, Do okna vložíme data: { "firstName": "Michael", "lastName": "Scott", "email": "scottym@mailto.com", "age": 151 }	Data se propíší
7, Stiskneme tlačítko Send	Objeví se status kód „400 Bad request“

9, Metoda POST

ID test case: tcpost.07

Název: Pokus o založení studenta s nesprávným věkem (0 let)

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- **Rozpětí pro věk je 1-150 let**
- firstName a lastName mohou obsahovat 3-50 znaků
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu POST	Objeví se nám žlutý nápis POST
2, Vložíme do vedlejšího volného pole tento odkaz: http://108.143.193.45:8080/api/v1/students/	Odkaz se propíše do řádku
3, O řádek níže zvolíme „Body“	Proklikneme se, pod se nám objeví další nabídka a okénko ke psaní
4, Z nabídky si vybereme „raw“ a formát „JSON“	Navolíme požadované formáty
5, Do okna vložíme data: { “firstName”: “Erik”, “lastName”: “Bjorn”, “email”: “eb@email.com”, “age”: 0 }	Data se propíší
6, Stiskneme tlačítko Send	Objeví se status kód „400 Bad request“

10, Metoda POST

ID test case: tcpost.08

Název: Pokus o založení studenta se správným věkem (hraniční hodnota 150 let)

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- **Všechny atributy se vkládají formou lowercase()**
- **Rozpětí pro věk je 1-150 let**
- firstName a lastName mohou obsahovat 3-50 znaků
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu POST	Objeví se nám žlutý nápis POST
2, Vložíme do vedlejšího volného pole tento odkaz: http://108.143.193.45:8080/api/v1/students/	Odkaz se propíše do řádku
3, O řádek níže zvolíme „Body“	Proklikneme se, pod se nám objeví další nabídka a okénko ke psaní
4, Z nabídky si vybereme „raw“ a formát „JSON“	Navolíme požadovaný formát
5, Do okna můžeme dle výběru vložit vždy jednoho studenta: { "firstName": "THOMAS", "lastName": "stone", "email": "sToNe@EmAiLy.CoM", "age": 150 }	Data se propíší do okna
6, Stiskneme tlačítko Send	A, Objeví se status kód „201 Created“, student dostane nové ID B, data se uloží formou lowercase()

11, Metoda POST

ID test case: tcpost.09

Název: Pokus o založení nového studenta s chybějícími údaji a záporným věkem

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- firstName a lastName mohou obsahovat 3-50 znaků
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu POST	Objeví se žlutý nápis POST
2, Vložíme do vedlejšího volného pole tento odkaz: http://108.143.193.45:8080/api/v1/students/	Odkaz se propše do řádku
3, O řádek níže zvolíme „Body“	Proklikneme se, pod se nám objeví další nabídka a okénko ke psaní
4, Z nabídky si vybereme „raw“ a formát „JSON“	Navolíme požadované formáty
5, Do okna vložíme následující data: { “firstName”: “”, “lastName”: “”, “email”: “”, “age”: -4 }	Data se propíší do okna
6, Stiskneme tlačítko Send	Objeví se status kód „400 bad request“

12, Metoda POST

ID test case: tcpost.10

Název: Pokus o založení studenta s emailem v nesprávném formátu

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- firstName a lastName mohou obsahovat 3-50 znaků
- **formát pro email je *@*.***

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu POST	Objeví se žlutý nápis POST
2, Vložíme do vedlejšího volného pole tento odkaz: http://108.143.193.45:8080/api/v1/students/	Odkaz se propše do řádku
3, O řádek níže zvolíme „Body“	Proklikneme se, pod se nám objeví další nabídka a okénko ke psaní
4, Z nabídky si vybereme „raw“ a formát „JSON“	Navolíme požadované formáty
5, Do okna vložíme následující data: { “firstName”: “Jim”, “lastName”: “Beam”, “email”: “@.aus, “age”: 70 }	Data se propíší do okna
6, Stiskneme tlačítko Send	Objeví se status kód „400 bad request“

13, Metoda DELETE

ID test case: tcdelete.01

Název: Smazání existujícího studenta s ID 1563

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- firstName a lastName mohou obsahovat 3-50 znaků
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu DELETE	Objeví se nám červený nápis DELETE
2, Do vedlejšího pole vložíme tento odkaz: http://108.143.193.45:8080/api/v1/students/1563	Vložený odkaz se propíše do pole
3, Stiskneme tlačítko Send	Objeví se status kód „200 OK“

14, Metoda DELETE

ID test case: tcdelete.02

Název: Pokus o smazání neexistujícího studenta s ID 42

Požadavky:

- Povinné atributy – id, firstName, lastName, email, age
- Všechny atributy se vkládají formou lowerCase()
- Rozpětí pro věk je 1-150 let
- firstName a lastName mohou obsahovat 3-50 znaků
- formát pro email je *@*.*

Postup:

KROKY	OČEKÁVANÝ VÝSLEDEK
1, Zvolíme metodu DELETE	Objeví se nám červený nápis DELETE
2, Do vedlejšího pole vložíme tento odkaz: http://108.143.193.45:8080/api/v1/students/42	Vložený odkaz se propíše do pole
3, Stiskneme tlačítko Send	Objeví se status kód „404 Not found“

EXEKUCE TESTŮ

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

1, Exekuce test case s ID: tcget.01

1, Zvolení metody GET – OK

2, Vložení odkazu – OK

3, Stisknutí tlačítka Send – Status Code 200 OK – OK

4, Zobrazení správných údajů – OK

The screenshot shows the Postman interface during the execution of a test case. The URL is set to `http://108.143.193.45:8080/api/v1/students/712`. The method is set to `GET` (1.). The response status is `200 OK` (3.), which is highlighted with a red oval. The response body (4.) is displayed in JSON format:

```
1. [GET]
2. http://108.143.193.45:8080/api/v1/students/712
3. 200 OK 131 ms 256 B
4.
{
  "id": 712,
  "firstName": "Armando",
  "lastName": "BIELY",
  "email": "armandbi@hismail.com",
  "age": 120
}
```

Závěr testu tcget.01: **PASS**

2, Exekuce test case s ID tcget.02

- 1, Zvolení metody GET – OK
- 2, Vložení odkazu – OK
- 3, Stisknutí send – očekávaný status kód „404 Not found“ – FAIL

The screenshot shows the Postman interface. At the top, the URL `http://108.143.193.45:8080/api/v1/students/50` is entered. Below it, a red box highlights the 'GET' method and the URL. A second red box highlights the 'Send' button. The 'Headers' tab is selected, showing '(6)' entries. The 'Body' tab is selected, showing 'Pretty' and 'Raw' options. The 'Test Results' tab is selected, showing a red circle around the response status: '500 Internal Server Error'. The response body is a JSON object:

```
1 "timestamp": "2024-08-14T12:40:24.141+00:00",
2 "status": 500,
3 "error": "Internal Server Error",
4 "message": "",
5 "path": "/api/v1/students/50"
```

- Ještě se dodatečně přesvědčím, že student v databázi opravdu neexistuje (pomocí SQL Workbench)

The screenshot shows the MySQL Workbench interface. A query is run: `Select * from student where id = 50;`. The results show a single row with all columns set to NULL:

id	age	email	first_name	last_name
NULL	NULL	NULL	NULL	NULL

Závěr testu tcget.02: FAIL – Postman nám místo status kódu 404 ukazuje kód 500 Internal Server Error (Zaneseno do bug report)

3, Exekuce test case s ID: tcpost.01

- 1, Zvolení metody POST – OK
- 2, Vložení odkazu – OK
- 3, Volba "Body" - OK
- 4, Volba formátu raw a JSON – OK
- 5, Zadání údajů – OK
- 6, Stisknutí tlačítka Send:
 - přidělení ID – OK
 - Status kód 201 – FAIL

The screenshot shows the Postman interface during a POST request to `http://108.143.193.45:8080/api/v1/students/`. The request details are as follows:

- 1. Method: POST
- 2. URL: `http://108.143.193.45:8080/api/v1/students/`
- 3. Body tab selected (highlighted with a red oval)
- 4. Content type: raw (highlighted with a red oval)
- 5. Request body content (highlighted with a red box):

```
1
2   ...
3     "firstName": "Bobby",
4     "lastName": "Stanley",
5     "email": "bs007@yamail.com",
6     "age": 1
```

- 6. Response status: 200 OK (highlighted with a red oval)

The response details are as follows:

- Body tab selected
- 5. Response status: 200 OK (highlighted with a red oval)
- 6. Response content (highlighted with a red box):

```
1
2   ...
3     "id": 1582,
4     "firstName": "Bobby",
5     "lastName": "STANLEY",
6     "email": "bs007@yamail.com",
7     "age": 1
```

- Ověření existence studenta pomocí SQLWorkbench – OK

The screenshot shows the SQLWorkbench interface displaying the results of a SELECT query:

```
1 • Select * from student where id = 1582;
```

The result grid shows the following data:

	id	age	email	first_name	last_name
1	1582	1	bs007@yamail.com	Bobby	STANLEY
2	NULL	NULL	NULL	NULL	NULL

Závěr testu tcpost.01 – PASS (student založen, chybný status kód již zanesen do bug report)

4, Exekuce test case s ID: tcpost.02

- 1, Zvolení metody POST - OK
- 2, Vložení odkazu – OK
- 3, Volba "Body" - OK
- 4, Volba raw a JSON – OK
- 5, Vložení údajů – OK
- 6, Stisknutí tlačítka Send – 400 Bad request – FAIL

HTTP <http://108.143.193.45:8080/api/v1/students/> Save

POST 1. <http://108.143.193.45:8080/api/v1/students/> 2.

Send

Params Authorization Headers (8) 3. Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary JSON 4.

1. "firstName": "Yi",
2. "lastName": "Yu",
3. "email": "iu@email.com",
4. "age": 29
5.

Body Cookies Headers (5) Test Results 6. 200 OK 447 ms 243 B Save Response

Pretty Raw Preview Visualize JSON

1. "id": 1558,
2. "firstName": "Yi",
3. "lastName": "Yu",
4. "email": "iu@email.com",
5. "age": 29
6.

- Následné ověření, že se student opravdu založil:

Select * from student where id = 1558;

	id	age	email	first_name	last_name
1	1558	29	iu@email.com	Yi	Yu
	HULL	HULL	HULL	HULL	HULL

Závěr testu tcpost.02 – **FAIL** – u bodu č. 6 vidíme, že student šel založit s firstName i lastName, které mají méně než 3 znaky (špatný status kód zde opomíjím, ten je zanesen do bug report, funguje u jakéhokoliv založení nového studenta)

5, Exekuce test case s ID: tcpost.03

- 1, Zvolení metody POST - OK
- 2, Vložení odkazu – OK
- 3, Volba "Body" - OK
- 4, Volba raw a JSON – OK
- 5, Vložení údajů – OK
- 6, Stisknutí tlačítka Send
 - student založen s novým ID – OK
 - status kód 201 - FAIL

The screenshot shows the Postman interface during the execution of a test case. The top bar displays the URL `http://108.143.193.45:8080/api/v1/students/`. The request method is set to `POST` (1.). The body tab is selected (3.). The content type is set to `JSON` (4.). The body payload (5.) contains the following JSON data:

```
1
2   ...
3     "firstName": "Yin",
4     "lastName": "Yun",
5     "email": "yy3@mailito.com",
6     "age": 31
```

The response (6.) shows a successful `200 OK` status with a response size of `245 B`. The response body (boxed) contains the newly created student's details:

```
1
2   ...
3     "id": 1557,
4     "firstName": "Yin",
5     "lastName": "YUN",
6     "email": "yy3@mailito.com",
7     "age": 31
```

- Ověření založení studenta:

The screenshot shows the MySQL Workbench interface with a query window containing the following SQL statement:

```
1 • Select * from student where id = 1557;
```

The results grid (Result Grid) shows one row of data:

id	age	email	first_name	last_name
1557	31	yy3@mailito.com	Yin	YUN

Závěr testu tcpost.03 – **PASS** – Student se nám opravdu založil, i když se špatným status kódem (ten už je zanesen v bug report)

6, Exekuce test case s ID: tcpost.04

- 1, Zvolení metody POST - OK
- 2, Vložení odkazu – OK
- 3, Volba "Body" - OK
- 4, Volba raw a JSON – OK
- 5, Vložení údajů – OK
- 6, Stisknutí tlačítka Send – 400 Bad request – FAIL

HTTP <http://108.143.193.45:8080/api/v1/students/> Save

POST 1. <http://108.143.193.45:8080/api/v1/students/> 2.

Params Authorization Headers (8) 3. Body 4. Tests Settings Cookies Beautify

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

1. 2. 3. 4. 5. 6.

Body Cookies Headers (5) Test Results 200 OK 186 ms 353 B Save Response

Pretty Raw Preview Visualize JSON

1. 2. 3. 4. 5. 6.

1. 2. 3. 4. 5. 6.

- Ověření založení studenta:

Select * from student where id = 1559;

id	age	email	first_name	last_name
1559	120	lakecity@yahoo.com	Chargoggagoggmanchauggagoggchaubunagungamauggaaaaaa	LLANFAIRPWLWGWNNGYLLGOGERYCHWYRNDROBWLLANTYSILIOGOGOGOCH
NULL	NULL	NULL	NULL	NULL

Závěr testu tcpost.04 – **FAIL** – Student se založil i přes vyšší počet znaků, než je uvedeno v požadavcích (3-50 znaků). Zaneseno do bug report.

7, Exekuce test case s ID: tcpost.05

1, Zvolení metody POST - OK

2, Vložení odkazu – OK

3, Volba "Body" - OK

4, Volba raw a JSON – OK

5, Vložení údajů – OK

6, Stisknutí tlačítka Send

- 201 Created – FAIL
- Nové ID – OK

The screenshot shows the Postman interface during the execution of a test case. The steps are numbered as follows:

1. Method: POST
2. URL: http://108.143.193.45:8080/api/v1/students/
3. Body tab selected.
4. Content type set to JSON.
5. Request body (JSON payload) containing student data.
6. Response status: 200 OK, 185 ms, 338 B.

The response body is shown in the screenshot:

```
1
2   ...
3     "id": 1560,
4     "firstName": "Bernadetteevangelinejacquelinemarylouisewilhelmina",
5     "lastName": "ALESSANDROFLORENTINOMONTGOMERYRUTHERFORDWASHINGTON",
6     "email": "ba50@email.com",
7     "age": 60
```

- Ověření založení studenta:

The screenshot shows the MySQL Workbench interface displaying the results of a query to verify the created student.

Query:

```
1 • Select * from student where id = 1560;
```

Result Grid:

id	age	email	first_name	last_name
1560	60	ba50@email.com	Bernadetteevangelinejacquelinemarylouisewilhelmina	ALESSANDROFLORENTINOMONTGOMERYRUTHERFORDWASHINGTON
HULL	HULL	HULL	HULL	HULL

Závěr testu tcpost.05 – **PASS** – Student se nám vytvořil (špatný status kód je známá chyba, již zanesena do bug report)

8, Exekuce test case s ID: tcpost.06

- 1, Zvolení metody POST - OK
- 2, Vložení odkazu – OK
- 3, Volba "Body" - OK
- 4, Volba raw a JSON – OK
- 5, Vložení údajů – OK
- 6, Stisknutí tlačítka Send – 400 Bad request – FAIL

The screenshot shows the Postman interface. Step 1 highlights the 'POST' method. Step 2 highlights the URL field. Step 3 highlights the 'Body' tab. Step 4 highlights the 'JSON' dropdown. Step 5 highlights the JSON payload in the body editor. Step 6 highlights the response status bar at the bottom.

- Ověření založení studenta:

The screenshot shows the MySQL Workbench interface with a query results grid. The query is 'Select * from student where id = 1561;'. The result grid displays the following data:

	id	age	email	first_name	last_name
1	1561	151	scottym@mailto.com	Michael	SCOTT
2	NULL	NULL	NULL	NULL	NULL

Závěr testu tcpost.06 – **FAIL** – Student se založil s věkem vyšším, než je zadáno v požadavcích. Zaneseno do bug report.

9, Exekuce test case s ID: tcpost.07

- 1, Zvolení metody POST - OK
- 2, Vložení odkazu – OK
- 3, Volba "Body" - OK
- 4, Volba raw a JSON – OK
- 5, Vložení údajů – OK
- 6, Stisknutí tlačítka Send – 400 Bad request – OK

The screenshot shows the Postman application interface. Step 1 highlights the method selection dropdown set to 'POST'. Step 2 highlights the URL input field containing 'http://108.143.193.45:8080/api/v1/students/'. Step 3 highlights the 'Body' tab in the top navigation bar. Step 4 highlights the 'raw' and 'JSON' buttons in the body settings. Step 5 highlights the JSON payload in the body editor, which contains the following data:

```
1
2   ...
3     "firstName": "Erik",
4     "lastName": "Bjorn",
5     "email": "eb@email.com"
6     "age": 0
```

Step 6 highlights the status code '400 Bad Request' in the bottom right corner of the response panel.

Závěr testu tcpost.07 - PASS

10, Exekuce test case s ID: tcpost.08

- 1, Zvolení metody POST - OK
- 2, Vložení odkazu – OK
- 3, Volba "Body" - OK
- 4, Volba raw a JSON – OK
- 5, Vložení údajů – OK
- 6, Stisknutí tlačítka Send
 - A. 201 Created – FAIL
Nové ID – PASS
 - B. Lowercase() – FAIL

The screenshot shows the Postman interface with the following steps highlighted:

1. Method: POST
2. URL: http://108.143.193.45:8080/api/v1/students/
3. Body tab selected (highlighted with a red oval)
4. Content type: raw (highlighted with a red oval)
5. Request body JSON (highlighted with a red box):

```
1
2   ...
3     "firstName": "THOMAS",
4     "lastName": "stone",
5     "email": "sToNe@EmAiLy.CoM",
6     "age": 150
```

6. Response status: 200 OK (highlighted with a red oval)

- Ověření založení studenta:

The screenshot shows the MySQL Workbench interface with the following results:

```
1 • Select * from student where id = 1580;
```

	id	age	email	first_name	last_name
-	1580	150	sToNe@EmAiLy.CoM	THOMAS	STONE
-	HULL	HULL	HULL	HULL	HULL

Závěr testu tcpost.08:

A, PASS - Student se založil s novým ID a správným věkem
(opomíjím špatný status kód – známá chyba, v bug report)

B, FAIL - funkce lowercase() nefunguje správně (zaneseno do
bug report)

11, Exekuce test case s ID: tcpost.09

- 1, Zvolení metody POST - OK
- 2, Vložení odkazu – OK
- 3, Volba "Body" - OK
- 4, Volba raw a JSON – OK
- 5, Vložení údajů – OK
- 6, Stisknutí tlačítka Send – 400 Bad request – FAIL

The screenshot shows the Postman interface during a POST request to `http://108.143.193.45:8080/api/v1/students/`. The request body is set to `JSON` (circled as 4). The body content (circled as 5) is:

```
1
2   ...
3     "firstName": "",
4     "lastName": "",
5     "email": "",
6     "age": -4
```

The response (circled as 6) is a 200 OK status with a response time of 184 ms and a size of 224 B.

- Ověření založení studenta:

The screenshot shows the MySQL Workbench interface with a query window containing the SQL statement:

```
1 • Select * from student where id = 1583;
```

The result grid displays the following data:

	id	age	email	first_name	last_name
1	1583	-4	NULL	NULL	NULL

Závěr testu tcpost.09 – **FAIL** – Student se založí i s chybějícími údaji a záporným věkem, zaneseno do bug report

12, Exekuce test case s ID: tcpost.10

- 1, Zvolení metody POST - OK
- 2, Vložení odkazu – OK
- 3, Volba "Body" - OK
- 4, Volba raw a JSON – OK
- 5, Vložení údajů – OK
- 6, Stisknutí tlačítka Send – 400 Bad request – FAIL

The screenshot shows the Postman interface. At the top, the URL is `http://108.143.193.45:8080/api/v1/students/`. The method is set to **POST** (1.). The body tab is selected (3.). The content type is set to **JSON** (4.). The body payload (5.) is a JSON object:

```
1
2   ...
3     "firstName": "Jim",
4     "lastName": "Beam",
5     "email": "@.aus",
6     "age": 70
```

The response (6.) shows a 200 OK status with the following JSON data:

```
1
2   ...
3     "id": 1581,
4     "firstName": "Jim",
5     "lastName": "BEAM",
6     "email": "@.aus",
7     "age": 70
```

- Ověření založení studenta:

The screenshot shows the MySQL Workbench interface. A query is run: `Select * from student where id = 1581;`. The result grid shows one row of data:

id	age	email	first_name	last_name
1581	70	@.aus	Jim	BEAM

Závěr testu tcpost.10 – FAIL – student se založil i se špatně zadáným emailem

13, Exekuce test case s ID: tcdelete.01

- Před provedením testu si ověřím existenci studenta s ID 1563:

The screenshot shows a MySQL Workbench interface. In the top query editor, the SQL command `Select * from student where id = 1563;` is entered. Below it, the results are displayed in a grid:

id	age	email	first_name	last_name
1563	16	DARI@mailto.com	RICKY	DALE
NULL	NULL	NULL	NULL	NULL

1, Zvolení metody DELETE – OK

2, Vložení odkazu – OK

3, Stisknutí tlačítka Send – status kód 200 OK – OK

The screenshot shows a Postman request configuration and its response. The method is set to `DELETE` (1.) and the URL is `http://108.143.193.45:8080/api/v1/students/1563` (2.). The response status is highlighted with a red oval (3.).

Body → Cookies → Headers (4) → Test Results

200 OK 126 ms 123 B

- Ověření smazání studenta z databáze – OK

The screenshot shows a MySQL Workbench interface. In the top query editor, the SQL command `Select * from student where id = 1563;` is entered. Below it, the results are displayed in a grid:

id	age	email	first_name	last_name
NULL	NULL	NULL	NULL	NULL

Závěr testu tcdelete.01 – **PASS**

13, Exekuce test case s ID: tcdelete.02

- Před provedením testu si ověřím, že student s ID 42 opravdu neexistuje:

The screenshot shows a MySQL Workbench interface. A query window contains the SQL command: `Select * from student where id = 42;`. The result grid below shows five columns: id, age, email, first_name, and last_name. All cells in the grid are empty (NULL), indicating no data was found for the specified ID.

- Zvolení metody DELETE - OK
- Vložení odkazu – OK
- Stisknutí tlačítka Send – status kód 404 Not found – FAIL

The screenshot shows a Postman request configuration. The method is set to **DELETE** (1.) and the URL is `http://108.143.193.45:8080/api/v1/students/42` (2.). The response section shows a red circle around the status bar, which displays **3.**, a 500 Internal Server Error, 103 ms, and 286 B. The response body is a JSON object with the following content:

```
1 "timestamp": "2024-08-14T15:01:40.691+00:00",
2 "status": 500,
3 "error": "Internal Server Error",
4 "message": "",
5 "path": "/api/v1/students/42"
```

Závěr testu tcdelete.02 – FAIL – Postman nám ukazuje místo očekávaného status kódu 404 kód 500 Internal Server Error (zaneseno do bug report)

BUG REPORT

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

1, BUG REPORT

ID: br.01

Název: Nefunkčnost požadavku lowercase()

Popis:

Data by se na základě uvedených požadavků měla ukládat formou lowercase(), což se bohužel neděje. Data zůstanou v takovém formátu, v jakém se zapíšou, kromě atributu lastName, který vždy udělá všechna písmena velká. Tento bug se vyskytuje napříč všemi exekucemi testů.

Přílohy:

Přikládám screeny z různých testů, kde je problém vidět:

- tcpost.08 – nejlepší příklad

The screenshot shows a POST request to `http://108.143.193.45:8080/api/v1/students/`. The request body is set to `raw` and `JSON`. The JSON payload is:

```
1. "firstName": "THOMAS",
2. "lastName": "stone",
3. "email": "sToNe@EmAiLy.CoM",
4. "age": 150
```

The response status is `200 OK`.

- tcpost.03

1. POST | 2. http://108.143.193.45:8080/api/v1/students/ | 3. Headers (8) | 4. Body | 5. Request Body JSON | 6. Response Status 200 OK

```

1
2   ...
3     "firstName": "Yin",
4     "lastName": "Yun",
5     "email": "yy3@mailito.com",
6     "age": 31
7

```

Body | Cookies | Headers (5) | Test Results | 200 OK | 409 ms | 245 B | Save Response

- Z databáze SQL je také vidět, že lowercase() nefunguje

1. Select * from student;

	id	age	email	first_name	last_name
344	21	pepaumackaw@gmail.com	Pepa		
345	34	john.doe@gmail.com	John	DOE	
347	18	alfredmail.com	alfred	HITCHCOCK	
349	80	alfred@mail.com	al	HITCHCOCK	
350	34	john.doe@gmail.com	Jo	DOE	
351	80	alfred@mail.com	al	HITCHCOCK	
352	17	john.doe@gmail.com	John	DOE	
353	18	john.doe@gmail.com	John	DOE	
354	18	alfred@mail.com	alfred	HITCHCOCK	
357	18	john.doe@gmail.com	John	DOE	
358	25	petrA-sova.cz	Petra	SOVA	
359	969	john.doe@gmail.com	John	DOE	
360	19	john.doe@.com	John	DOE	
361	17	petrA@sova.cz	Petra	SOVA	
362	25	petrA@sova.cz	Pe	SOVA	
363	34	john.doe@gmail.com	John	DO	

student 20 | Apply | Revert | Result Grid | Form Editor | Field Types | Query Stats

2, BUG REPORT

ID: br.02

Název: Nesprávný status kód při zakládání studentů

Popis:

Správně by se při založení studenta do databáze měl objevit status kód „201 Created“, to se ovšem neděje a objevuje se status kód „200 OK“.

Přílohy:

- tcpost.01

The screenshot shows a POST request in Postman. The URL is `http://108.143.193.45:8080/api/v1/students/`. The request body is a JSON object:

```
1. "firstName": "Bobby",  
2. "lastName": "Stanley",  
3. "email": "bs007@yamail.com",  
4. "age": 1
```

The response status is 200 OK, with a response time of 186 ms and a response size of 251 B. The response body is:

```
1. {  
2.   "id": 1582,  
3.   "firstName": "Bobby",  
4.   "lastName": "STANLEY",  
5.   "email": "bs007@yamail.com",  
6.   "age": 1  
7. }
```

Annotations:

1. POST button
2. Request URL
3. Body tab
4. JSON dropdown
5. Request body JSON content
6. Response body JSON content
7. Response status (200 OK)

- tcpost.06

The screenshot shows the Postman interface with the following numbered steps:

1. Method: POST | URL: http://108.143.193.45:8080/api/v1/students/
2. Pre-request Script: 2: http://108.143.193.45:8080/api/v1/students/
3. Body tab selected.
4. Content type: JSON selected.
5. Request body (JSON):

```

1   ...
2   ...
3   ...
4   ...
5   ...
6   ...
    "firstName": "Michael",
    "lastName": "Scott",
    "email": "scottym@mailto.com",
    "age": 151
  
```

6. Response status: 200 OK | 186 ms | 255 B

Below the request, the response is displayed in Pretty mode:

```

1   ...
2   ...
3   ...
4   ...
5   ...
6   ...
7   ...
      "id": 1561,
      "firstName": "Michael",
      "lastName": "SCOTT",
      "email": "scottym@mailto.com",
      "age": 151
    
```

3, BUG REPORT

ID: br.03

Název: Místo status kódu „404 Not found“ se objevuje kód „500 Internal Server Error“

Popis:

U scénářů, kde v očekávaném výsledku měl vyjít status kód „404 Not found“, se následně při jejich exekuci objevuje status kód „500 Internal Server Error“, databáze si s tím neumí poradit.

Přílohy:

- tcget.02

The screenshot shows a POSTMAN interface with the following details:

- Method:** GET (highlighted in red, labeled 1.)
- URL:** http://108.143.193.45:8080/api/v1/students/50 (highlighted in red, labeled 2.)
- Headers:** (6)
- Body:** (empty)
- Pre-request Script:** (empty)
- Tests:** (empty)
- Settings:** (empty)
- Cookies:** (empty)
- Query Params:** (empty table)
- Body:** (selected tab)
- Cookies:** (empty)
- Headers:** (4)
- Test Results:** (empty)
- Status Bar:** 3. 500 Internal Server Error | 139 ms | 286 B | Save Response
- Bottom Panel:** Pretty (selected), Raw, Preview, Visualize, JSON, Copy, Find, Save

A red circle highlights the "500 Internal Server Error" status message in the status bar at the bottom right.

```
1 {"timestamp": "2024-08-14T12:40:24.141+00:00",  
2 "status": 500,  
3 "error": "Internal Server Error",  
4 "message": "",  
5 "path": "/api/v1/students/50"
```

- tcdelete.02

The screenshot shows the Postman interface with the following details:

- Method:** **DELETE** (1.)
- URL:** <http://108.143.193.45:8080/api/v1/students/42> (2.)
- Send** button (3.)
- Headers:** (6)
- Body:** (Empty)
- Tests:** (Empty)
- Settings:** (Empty)
- Cookies:** (Empty)
- Query Params:** (Empty)
- Body:** (Pretty) (4.)
- Headers:** (4.)
- Test Results:** (Empty)
- Response Headers:** (500 Internal Server Error, 103 ms, 286 B) (3.)
- Save Response** button (3.)
- Body Content:**

```
1 "timestamp": "2024-08-14T15:01:40.691+00:00",
2 "status": 500,
3 "error": "Internal Server Error",
4 "message": "",
5 "path": "/api/v1/students/42"
```

4, BUG REPORT

ID: br.04

Název: Nefunkčnost požadavku u atributu „age“

Popis:

Po otestování hodnot pro zadávání věku vyšlo najevo, že můžeme zadávat do databáze i studenty, kterým je více než 150 let, tedy jdou věkem přes horní povolenou hranici. U exekuce scénáře tcpost.07 byla zadána hodnota 0 a student se nezaložil. Avšak později, při exekuci tcpost.09 byla zadána záporná hodnota věku -4 a student se i přesto založil.

Přílohy:

- tcpost.06

The screenshot shows a POST request to `http://108.143.193.45:8080/api/v1/students/`. The request body is JSON, containing the following data:

```
1. "firstName": "Michael",
2. "lastName": "Scott",
3. "email": "scottym@mailto.com",
4. "age": 151
```

The response status is 200 OK, indicating the student was successfully created despite the age being above the limit.

1 • `Select * from student where id = 1561;`

id	age	email	first_name	last_name
1561	151	scottym@mailto.com	Michael	SCOTT
NULL	NULL	NULL	NULL	NULL

- tcpost.09

1. `POST http://108.143.193.45:8080/api/v1/students/`

2. `http://108.143.193.45:8080/api/v1/students/`

3. Body

4. raw JSON

5. `{
 "firstName": "",
 "lastName": "",
 "email": "",
 "age": -4
}`

6. `200 OK 184 ms 224 B`

7. `{
 "id": 1583,
 "firstName": "",
 "lastName": "",
 "email": "",
 "age": -4
}`

1 • `Select * from student where id = 1583;`

id	age	email	first_name	last_name
1583	-4	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL

5, BUG REPORT

ID: br.05

Název: Studenti jdou založit i s chybějícími atributy

Popis:

V požadavcích je uvedeno, že všechny uvedené atributy (id, firstName, lastName, email, age) jsou při zakládání nového studenta povinné. Po otestování tohoto požadavku vyšlo najevo, že student jde založit i s chybějícími atributy.

Přílohy:

- tcpost.09

The screenshot shows a POST request in Postman. The URL is `http://108.143.193.45:8080/api/v1/students/`. The request body is a JSON object with the following content:

```
1. "firstName": "",  
2. "lastName": "",  
3. "email": "",  
4. "age": -4
```

The body tab is selected (3.). The content type is set to `JSON` (4.). The response status is `200 OK` (6.).

The screenshot shows a MySQL query results window. The query is `Select * from student where id = 1583;`. The result grid shows one row with the following data:

id	age	email	first_name	last_name
1583	-4	NULL	NULL	NULL

6, BUG REPORT

ID: br.06

Název: Nefunguje daný formát atributu „email“

Popis:

V požadavcích je uveden formát, ve kterém má být zadáván email studenta (*@*.*)
ten však nefunguje a email lze uložit i v nesprávné formě.

Přílohy:

- tcpost.10

The screenshot shows a POST request to `http://108.143.193.45:8080/api/v1/students/`. The request body is set to `JSON` (4.) and contains the following JSON payload (5.):

```
1. {  
2.   "firstName": "Jim",  
3.   "lastName": "Beam",  
4.   "email": "@.aus",  
5.   "age": 70  
6. }
```

The response (6.) is a 200 OK status with a response time of 194 ms and a response size of 236 B.

The screenshot shows a MySQL query result grid. The query is:

```
1. Select * from student where id = 1581;
```

The result grid shows the following data:

id	age	email	first_name	last_name
1581	70	@.aus	Jim	BEAM
HULL	NULL	NULL	NULL	NULL

7, BUG REPORT

ID: br.07

Název: Nefungování požadavku pro rozpětí znaků ve firstName a lastName

Popis:

V požadavcích je uvedeno, že „firstName a lastName mohou obsahovat 3-50 znaků“. To však nefunguje, při exekucích se ukázalo, že jména jdou ukládat jak se 2 znaky, ale i s více než 50.

Přílohy:

- tcpost.02

The screenshot shows a POST request to `http://108.143.193.45:8080/api/v1/students/`. The request body is a JSON object with the following fields:

```
1. POST 2. http://108.143.193.45:8080/api/v1/students/
3. Body (highlighted with a red circle)
4. raw (highlighted with a red circle)
5. JSON (highlighted with a red circle)
6. Response status: 200 OK, 447 ms, 243 B
```

The response body is also a JSON object with the same fields as the request body, indicating successful creation of the student record.

The screenshot shows the results of a MySQL query:

```
1. Select * from student where id = 1558;
```

The result grid displays the following data:

	id	age	email	first_name	last_name
	1558	29	iu@email.com	Yi	YU
	HULL	HULL	HULL	HULL	HULL

- tcpost.04

The screenshot shows the Postman interface. A red box highlights the URL field (1.) containing "POST" and "http://108.143.193.45:8080/api/v1/students/". Another red box highlights the Headers tab (3.) with "Body" selected. A third red box highlights the "raw" radio button (4.) under the Body type dropdown. A fourth red box highlights the JSON response body (5.) which contains a student object with fields: id, firstName, lastName, email, and age. A fifth red box highlights the status bar at the bottom right (6.) showing "200 OK", "186 ms", and "353 B".

```

1. POST http://108.143.193.45:8080/api/v1/students/
2. http://108.143.193.45:8080/api/v1/students/
3. Headers (8) Body
4. raw
5.
{
  "id": 1559,
  "firstName": "Chargoggagoggmanchauggagoggchaubunagungamauggaaaaaa",
  "lastName": "LLANFAIRPWLLGWYNGYLLGOGERYCHWYRNDROBWLLLANTYSILIOGOGOGOCH",
  "email": "lakecity@yahoo.com",
  "age": 120
}
6. 200 OK 186 ms 353 B
    
```

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query (1.) is "Select * from student where id = 1559;". The result grid (2.) displays a single row of data:

id	age	email	first_name	last_name
1559	120	lakecity@yahoo.com	Chargoggagoggmanchauggagoggchaubunagun...	LLANFAIRPWLLGWYNGYLLGOGERYCHWYR...