

FINÁLNÍ PROJEKT

č.1



Autor: Alice Kočířová
Datum: 15.5.202

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	5
BUG REPORT	5

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 Host: aws.connect.psdb.cloud Username: 68q0llcbuyijdt5mzq2 Password: pscale_pw_zxfenAXfSYx9loe9PCS1snuGKr Fzw2J84BMIVZI96o
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.

Testovací scénář č. 1 metoda GET

Testovací scénář č. 2 metoda POST

Testovací scénář č. 3 metoda DELETE

EXEKUCE TESTŮ

1. METODA GET

1. Získání informací o existujícím studentovi pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:800api/v1/students/2913. Odeslání požadavku GET4. Analýza status kódu5. Ověření správnosti informací	<ul style="list-style-type: none">- Status kód je 200 OK- Odpověď obsahuje informace o studentovi ID 291- Informace o studentovi odpovídají

```
Pretty  Raw  Preview  Visualize  JSON  ⌵  ⌵
1  {
2    "id": 291,
3    "firstName": "Adel",
4    "lastName": "SVIDRON",
5    "email": "adelsvidron@mail.com",
6    "age": 11
7  }
```

2. Získání informací o existujícím studentovi pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/3293. Odeslání požadavku GET4. Analýza status kódu5. Ověření správnosti informací	<ul style="list-style-type: none">- Status kód je 200 OK- Odpověď obsahuje informace o studentovi ID 329- Informace o studentovi odpovídají

```
1 {
2   "id": 329,
3   "firstName": "Daenerys",
4   "lastName": "TARGARYEN",
5   "email": "danny.khaleesi@mail.com",
6   "age": 24
7 }
```

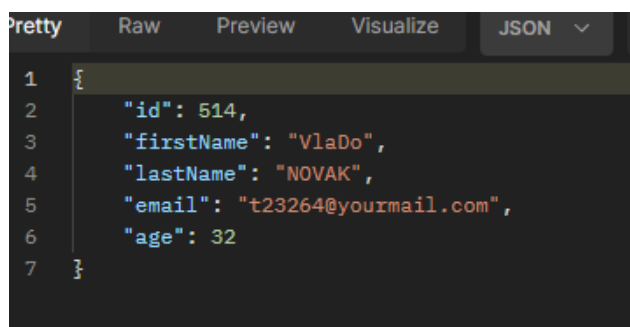
3. Získání informací o existujícím studentovi pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/5093. Odeslání požadavku GET4. Analýza status kódu5. Ověření správnosti informací	<ul style="list-style-type: none">- Status kód je 200 OK- Odpověď obsahuje informace o studentovi ID 509- Informace o studentovi odpovídají

```
1 {
2   "id": 509,
3   "firstName": "Alžběta",
4   "lastName": "DIVNÁ",
5   "email": "ja_divna@yourmail.com",
6   "age": 89
7 }
```

4. Získání informací o existujícím studentovi pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/5143. Odeslání požadavku GET4. Analýza status kódu5. Ověření správnosti informací	<ul style="list-style-type: none">- Status kód je 200 OK- Odpověď obsahuje informace o studentovi ID 514- Informace o studentovi odpovídají



```
Pretty  Raw  Preview  Visualize  JSON  v
1  {
2    "id": 514,
3    "firstName": "VlaDo",
4    "lastName": "NOVAK",
5    "email": "t23264@yourmail.com",
6    "age": 32
7  }
```

5. Získání informací o studentovi s chybně uvedeným ID pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/20.23. Odeslání požadavku GET4. Analýza status kódu	<ul style="list-style-type: none">- Status kód je 400 Bad Request <pre>{ "timestamp": "2024-05-05T15:55:04.645+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/20.2" }</pre>

6. Získání informací o studentovi podle jména pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/Alice3. Odeslání požadavku GET4. Analýza status kódu	<p>- Status kód je 400 Bad Request</p> <pre>{ "timestamp": "2024-05-05T15:55:51.677+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/alice"}</pre>

7. Získání informací o studentovi s použitím jiného znaku pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/&3. Odeslání požadavku GET4. Analýza status kódu	<p>- Status kód je 400 Bad Request</p> <pre>{ "timestamp": "2024-05-05T15:57:22.594+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/&"}</pre>

Poznámka: platí i pro ostatní znaky např.: : , ! + < - @ =

8. Získání informací o neexistujícím studentovi pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/10803. Odeslání požadavku GET4. Analýza status kódu	<p>- Status kód je 404 Not found -Požadovaný student s ID 1080 nebyl nalezen na serveru</p>

9. Získání informací o studentovi s použitím znaku ? pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/ ?3. Odeslání požadavku GET4. Analýza status kódu	<p>- Status kód je 400 Bad Request</p> <pre>{ "timestamp": "2024-05-05T15:24:48.708+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/?" }</pre>

10. Získání informací o studentovi s mezerou pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/6663. Odeslání požadavku GET4. Analýza status kódu	<p>- Status kód je 400 Bad Request</p> <pre>{ "timestamp": "2024-05-05T15:24:48.708+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/ 666" }</pre>

11. Získání informací o studentovi id s pomlčkou pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/-8933. Odeslání požadavku GET4. Analýza status kódu	<p>- Status kód je 400 Bad Request</p> <pre>{ "timestamp": "2024-05-05T15:24:48.708+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/-893" }</pre>

12. Získání informací o studentovi s použitím znaku % pomocí metody GET:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/%3. Odeslání požadavku GET4. Analýza status kódu	<p>- Status kód je 400 Bad Request</p> <pre>{ "timestamp": "2024-05-05T15:24:48.708+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/%" }</pre>

2. METODA POST

1. Založení nového studenta pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Alice", "lastName": "Kocirova", "age": 32, "email": "kocirova@email.cz" }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<p>- Status 200 OK</p> <ul style="list-style-type: none">• id: 980• age: 32• email: kocirova@email.cz• first_name: Alice• last_name: Kocirova

2. Založení nového studenta pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Marge", "lastName": "HOMEROVA", "age": 40, "email": "MARGE@email.cz" }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 200 OK• id: 981• age:40• email: marge@email.cz• first_name: Marge• last_name: Homerova

3. Založení nového studenta pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "OLINA", "lastName": "OILOVA", "age": 48, "email": "olivova@email.cz" }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 200 OK• id: 983• age:48• email: olivova@email.cz• first_name: Olina• last_name: Oilova

4. Založení nového studenta s nevyplněným firstName, lastName pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": " ", "lastName": " ", "age": 20, "email": "noname@email.cz" }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

5. Založení nového studenta s nevyplněným emailem pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Shady ", "lastName": "Slim ", "age": 35, "email": " " }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

6. Založení nového duplicitního studenta pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Kurt", "lastName": "Cobain", "age": 27, "email": "nirvana@gmail.com " }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 409 Conflict

7. Založení nového studenta s nevyplněným emailem pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "John", "lastName": "Growl", "age": 56, "email": " " }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

8. Založení nového studenta s nevyplněným věkem pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Amelie", "lastName": "Lens", "age": , "email": "techno@email.cz " }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

9. Založení nového studenta s vyplněným věkem písmeny pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Amelie", "lastName": "Lens", "age": "dvacet", "email": "techno@email.cz " }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

10. Založení nového studenta s neplatnými hodnotami ve firstName a lastName pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "@!123", "lastName": "%=", "age": 18, "email": "zavinac@procento.cz " }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

11. Založení nového studenta s neplatnými údaji v emailu pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Charlota", "lastName": "Devit", "age": 18, "email": "%blabla" }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

12. Založení nového studenta s překročením limitu velikosti dat pomocí metody POST:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód: { "firstName": "nesmyslnedlouhejmenoktereby nemelobytulozeneblablablablablablablablab lablablablablablablablablablablablablablab raddokoladokoladokoladokola ", "lastName": "Velmidlouhejmeno", "age": 20, "email": "dlouhejmeno@gmail.com" }5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 413 Payload Too Large

13. Založení nového studenta s vložením https odkazu pomocí metody POST:





Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód: { "firstName": "https://www.google.cz/", "lastName": "odkaz", "age": 25, "email": "google@gmail.cz" }5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 403 Forbidden

3. METODA DELETE

1. Smazání existujícího studenta s ID 980 pomocí metody DELETE:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/9803. Odeslání požadavku DELETE4. Analýza status kódu5. Ověření dat v databázi	<ul style="list-style-type: none">- status kód 200 OK- v databázi je student smazán


```
1 • SELECT * FROM student WHERE id=980;
```

Result Grid				
Filter Rows: <input type="text"/>				
Edit:    Export/Import: 				
id	age	email	first_name	last_name
NULL	NULL	NULL	NULL	NULL

2. Smazání existujícího studenta s ID 981 pomocí metody DELETE:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/9813. Odeslání požadavku DELETE4. Analýza status kódu5. Ověření dat v databázi	<ul style="list-style-type: none">- status kód 200 OK- v databázi je student smazán






```
1 • SELECT * FROM student WHERE id=981;
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit:   					
Export/Import: 					
id	age	email	first_name	last_name	
NULL	NULL	NULL	NULL	NULL	

3. Smazání existujícího studenta s ID 983 pomocí metody DELETE:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/9833. Odeslání požadavku DELETE4. Analýza status kódu5. Ověření dat v databázi	<ul style="list-style-type: none">- status kód 200 OK- v databázi je student smazán

```
1 • SELECT * FROM student WHERE id=983;
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit:   					
Export/Import: 					
id	age	email	first_name	last_name	
NULL	NULL	NULL	NULL	NULL	

4. Smazání neexistujícího studenta s ID 70 pomocí metody DELETE:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/703. Odeslání požadavku DELETE4. Analýza status kódu	<ul style="list-style-type: none">- status kód 404 Not Found

5. Smazání neexistujícího studenta s ID 5 pomocí metody DELETE:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/53. Odeslání požadavku DELETE4. Analýza status kódu	<ul style="list-style-type: none">- status kód 404 Not Found

6. Smazání studenta s ID ? pomocí metody DELETE:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/?3. Odeslání požadavku DELETE4. Analýza status kódu	<ul style="list-style-type: none">- status kód 400 Bad Request

7. Smazání studenta s ID ! pomocí metody DELETE:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/!3. Odeslání požadavku DELETE4. Analýza status kódu	<ul style="list-style-type: none">- status kód 400 Bad Request

Poznámka: platí i pro jiné znaky např. + * “ > ;

```
{
  "timestamp": "2024-05-15T13:39:03.902+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "",
  "path": "/api/v1/students/!"
}
```

8. Smazání studenta podle jména Kurt pomocí metody DELETE:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/Kurt3. Odeslání požadavku DELETE4. Analýza status kódu	<ul style="list-style-type: none">- status kód 400 Bad Request

Poznámka: Platí i pro ostatní jména nebo písmena

```
{
  "timestamp": "2024-05-15T13:47:00.545+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "",
  "path": "/api/v1/students/Kurt"
}
```

9. Smazání studenta s nevyplněným ID pomocí metody DELETE:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Odeslání požadavku DELETE4. Analýza status kódu	<ul style="list-style-type: none">- status kód 405 Method Not Allowed

10. Smazání studenta podle % pomocí metody DELETE:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">5. Otevření programu Postman6. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/7. Odeslání požadavku DELETE8. Analýza status kódu	<ul style="list-style-type: none">- status kód 400 Bad Request

BUG REPORT

1. METODA GET:

Na základě provedených scénářů jsem objevila chyby u testů č.: 8, 9, 10, 11, 12, viz níže:

8. Získání informací o neexistujícím studentovi pomocí metody GET:

Abstract: Program vypíše chybný status kód:

Kroky:	Očekávaný výsledek:
1. Otevření programu Postman 2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/1080 3. Odeslání požadavku GET 4. Analýza status kódu	- Status kód je 404 Not found -Požadovaný student s ID 1080 nebyl nalezen na serveru

Skutečný výsledek:
- Status kód je 500 Internal Server Error <pre>{ "timestamp": "2024-05-05T16:12:13.871+00:00", "status": 500, "error": "Internal Server Error", "message": "", "path": "/api/v1/students/1080" }</pre>

9. Získání informací o studentovi se znakem ? pomocí metody GET:

Abstract: Program vypíše všechny záznamy studentů

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/ ?3. Odeslání požadavku GET4. Analýza status kódu	<ul style="list-style-type: none">- Status kód je 400 Bad Request <pre>{ "timestamp": "2024-05-05T15:24:48.708+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/?" }</pre>

Skutečný výsledek:

- Status kód je 200 OK
- Program vypíše všechny studenty
- Celkem 3432 řádků

```
1  [  
2    {  
3      "id": 175,  
4      "firstName": "",  
5      "lastName": "SENNA",  
6      "email": "ayrton@senna.com",  
7      "age": 34  
8    },  
9  ]
```

```
Pretty  Raw  Preview  Visualize  JSON  ↕  
3425    {  
3426      "id": 926,  
3427      "firstName": "Dana",  
3428      "lastName": "SCULLY",  
3429      "email": "spooky@xfiles.com",  
3430      "age": 56  
3431    }  
3432  ]
```

Poznámka: platí i pro znaky ; /

10. Získání informací o studentovi s mezerou pomocí metody GET:

Abstract: Program vypíše chybný status kód

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/6663. Odeslání požadavku GET4. Analýza status kódu	<p>- Status kód je 400 Bad Request</p> <pre>{ "timestamp": "2024-05-05T15:24:48.708+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/ 666" }</pre>

Skutečný výsledek:

```
firstName": "Armando",  
  "lastName": "BIELY",  
  "email": "armandbi@hismail.com",  
  "age": 24  
}
```

11. Získání informací o studentovi id s pomlčkou pomocí metody GET:

Abstract: Program vypíše chybný status kód:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/-8933. Odeslání požadavku GET4. Analýza status kódu	<p>- Status kód je 400 Bad Request</p> <pre>{ "timestamp": "2024-05-05T15:24:48.708+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/-893" }</pre>

Skutečný výsledek:

- status kód 500Internal Server Error

```
{
  "timestamp": "2024-05-05T16:26:33.404+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "message": "",
  "path": "/api/v1/students/-893"
}
```

12. Získání informací o studentovi s použitím znaku % pomocí metody GET:

Abstract: Program vypíše chybný status kód:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového GET požadavku s URL http://108.143.193.45:8080/api/v1/students/%3. Odeslání požadavku GET4. Analýza status kódu	<p>- Status kód je 400 Bad Request</p> <pre>{ "timestamp": "2024-05-05T15:24:48.708+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/%" }</pre>

Skutečný výsledek:

- Status kód je 400 Bad Request
- tělo odpovědi nemá správný formát a obsahuje HTML stránku s nadpisem "HTTP Status 400 – Bad Request"

```
<!doctype html>
<html lang="en">

<head>
  <title>HTTP Status 400 - Bad Request</title>
  <style type="text/css">
    body {
      font-family: Tahoma, Arial, sans-serif;
    }

    h1,
    h2,
```



```
h3,
b {
    color: white;
    background-color: #525D76;
}

h1 {
    font-size: 22px;
}

h2 {
    font-size: 16px;
}

h3 {
    font-size: 14px;
}

p {
    font-size: 12px;
}

a {
    color: black;
}

.line {
    height: 1px;
    background-color: #525D76;
    border: none;
}

</style>
</head>

<body>
    <h1>HTTP Status 400 - Bad Request</h1>
</body>

</html>
```

2. METODA POST:

Na základě provedených scénářů jsem objevila chyby u testů č.: 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, viz níže:

1. Založení nového studenta pomocí metody POST:

Abstract: Data v lastName se neuloží podle pravidla lowerCase

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Alice", "lastName": "Kocirova", "age": 32, "email": "kocirova@email.cz" }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 200 OK• id: 980• age: 32• email: kocirova@email.cz• first_name: Alice• last_name: Kocirova

Skutečný výsledek:

```
{
  "id": 980,
  "firstName": "Alice",
  "lastName": "KOCIROVA",
  "email": "kocirova@email.cz",
  "age": 32
}
```

	id ▲	age	email	first_name	last_name
▶	980	32	kocirova@email.cz	Alice	KOCIROVA
*	NULL	NULL	NULL	NULL	NULL

2. Založení nového studenta pomocí metody POST:

Abstract: Data v emailu se neuloží podle pravidla lowerCase:

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Marge", "lastName": "HOMEROVA", "age": 40, "email": "MARGE@email.cz" }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 200 OK• id: 981• age: 40• email: marge@email.cz• first_name: Marge• last_name: Homerova

Skutečný výsledek:

Status 200 OK

```
{
  "id": 981,
  "firstName": "Marge",
  "lastName": "HOMEROVA",
  "email": "MARGE@email.cz",
  "age": 40
}
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit:					
Export/Import:					
Wrap Cell Contents: <input type="checkbox"/>					
	id	age	email	first_name	last_name
▶	981	40	MARGE@email.cz	Marge	HOMEROVA
*	NULL	NULL	NULL	NULL	NULL

3. Založení nového studenta pomocí metody POST:

Abstract: Data v firstName se neuloží podle pravidla LowerCase

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">Otevření programu PostmanOtevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/Vyberu záložku Body, kliknu na RAW a nastavím formát JSONVypíši kód:<pre>{ "firstName": "OLINA", "lastName": "OILOVA", "age": 48, "email": "olivova@email.cz" }</pre>Odeslání požadavku POSTAnalýza status kóduKontrola informací s databází	<ul style="list-style-type: none">- Status 200 OK• id: 983• age: 48• email: olivova@email.cz• first_name: Olina• last_name: Oilova

Skutečný výsledek:

status 200 OK

```
{  
  "id": 983,  
  "firstName": "OLINA",  
  "lastName": "OILOVA",  
  "email": "olivova@email.cz",  
  "age": 48  
}
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit: Export/Import:					
	id	age	email	first_name	last_name
▶	983	48	olivova@email.cz	OLINA	OILOVA
*	NULL	NULL	NULL	NULL	NULL

4. Založení nového studenta s nevyplněným firstName, lastName pomocí metody POST:

Abstract: program nevypíše chybovou hlášku a uloží záznam

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": " ", "lastName": " ", "age": 20, "email": "noname@email.cz" }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

Skutečný výsledek:

Status 200 OK

```
{
  "id": 984,
  "firstName": " ",
  "lastName": " ",
  "email": "noname@email.cz",
  "age": 20
}
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit: Export/Import:					
	id	age	email	first_name	last_name
▶	984	20	noname@email.cz		
*	NULL	NULL	NULL	NULL	NULL

5. Založení nového studenta s nevyplněným emailem pomocí metody POST:

Abstract: Program nevypíše chybovou hlášku a záznam uloží

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Shady ", "lastName": "Slim ", "age": 35, "email": " " }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

Skutečný výsledek:
Status 200 OK: <pre>{ "id": 985, "firstName": "Shady", "lastName": "SLIM", "email": "", "age": 35 }</pre>

Result Grid					
Filter Rows:					
	id	age	email	first_name	last_name
▶	985	35		Shady	SLIM
*	NULL	NULL	NULL	NULL	NULL

6. Založení nového duplicitního studenta pomocí metody POST:

Abstract: Program uloží duplicitní záznam

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Kurt", "lastName": "Cobain", "age": 27, "email": "nirvana@gmail.com " }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 409 Conflict

Skutečný výsledek:

Status 200 OK

```
{
  "id": 987,
  "firstName": "Kurt",
  "lastName": "COBAIN",
  "email": "nirvana@gmail.com",
  "age": 27
}
```

Result Grid					
Filter Rows:					
	id	age	email	first_name	last_name
▶	986	27	nirvana@gmail.com	Kurt	COBAIN
*	NULL	NULL	NULL	NULL	NULL

Result Grid					
Filter Rows:			Edit:		
	id	age	email	first_name	last_name
▶	987	27	nirvana@gmail.com	Kurt	COBAIN
*	NULL	NULL	NULL	NULL	NULL

7. Založení nového studenta s nevyplněným emailem pomocí metody POST:

Abstract: Program uloží záznam o studentovi bez vyplněného emailu

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none"> 1. Otevření programu Postman 2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/ 3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON 4. Vypíši kód: <pre>{ "firstName": "John", "lastName": "Growl", "age": 56, "email": "" }</pre> 5. Odeslání požadavku POST 6. Analýza status kódu 7. Kontrola informací s databází 	<ul style="list-style-type: none"> - Status 400 Bad Request

Skutečný výsledek:

Status 200 OK

```
{
  "id": 988,
  "firstName": "John",
  "lastName": "GROWL",
  "email": "",
  "age": 56
}
```

Result Grid					
Filter Rows:			Edit:		
	id	age	email	first_name	last_name
▶	988	56		John	GROWL
*	NULL	NULL	NULL	NULL	NULL

10. Založení nového studenta s neplatnými hodnotami ve firstName a lastName pomocí metody POST:

Abstract: Program uloží záznam o studentovi s neplatnými hodnotami ve firstName a lastName

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "@!123", "lastName": "%="; ", "age": 18, "email": "zavinac@procento.cz " }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

Skutečný výsledek:

Status 200 OK

```
{
  "id": 989,
  "firstName": "@!123",
  "lastName": "%="; ",
  "email": "zavinac@procento.cz",
  "age": 18
}
```

Result Grid					
Filter Rows:					
	id	age	email	first_name	last_name
▶	989	18	zavinac@procento.cz	@!123	%=;
*	NULL	NULL	NULL	NULL	NULL

11. Založení nového studenta s neplatnými údaji v emailu pomocí metody POST:

Abstract: Program uloží záznam o studentovi s neplatnými hodnotami v emailu

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON4. Vypíši kód:<pre>{ "firstName": "Charlota", "lastName": "Devit", "age": 18, "email": "%blabla" }</pre>5. Odeslání požadavku POST6. Analýza status kódu7. Kontrola informací s databází	<ul style="list-style-type: none">- Status 400 Bad Request

Skutečný výsledek:
Status 200 OK <pre>{ "id": 990, "firstName": "Charlota", "lastName": "DEVIT", "email": "%blabla", "age": 18 }</pre>

Result Grid Filter Rows: <input type="text"/> Edit: Export/Import: Wrap Cell Cc					
	id	age	email	first_name	last_name
▶	990	18	%blabla	Charlota	DEVIT
*	NULL	NULL	NULL	NULL	NULL

Abstract: Program uloží záznam o studentovi s daty překračující limit

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none"> 1. Otevření programu Postman 2. Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/ 3. Vyberu záložku Body, kliknu na RAW a nastavím formát JSON 4. Vypíši kód: <pre>{ "firstName":"nesmyslnedlouhejmenoktereby nemelobytulozeneblablablablablablablab lablablablablablablablablablablablablab raddokoladokoladokoladokola ", "lastName":"Velmidlouhejmeno", "age":20, "email":"dlouhejmeno@gmail.com" }</pre> 5. Odeslání požadavku POST 6. Analýza status kódu 7. Kontrola informací s databází 	<ul style="list-style-type: none"> - Status 413 Payload Too Large

```
Skutečný výsledek:  
  
Status 200 OK  
{  
    "id": 991,  
    "firstName":  
"nesmyslnedlouhejmenokterebynemelobytulozeneblablablablablablablablablablablablablablablablablablablsblslvldlsatakdaleporaddokoladokoladokoladokola",  
    "lastName": "VELMIDLOUHEJMENO",  
    "email": "dlouhejmeno@gmail.cz",  
    "age": 20  
}
```

Result Grid						Filter Rows:	Edit:	Export/Import:	Wrap
	id	age	email	first_name	last_name				
▶	991	20	dlouhejmeno@gmail.cz	nesmyslnedlouhejme...	VELMIDLOUHEJMENO				
*	NULL	NULL	NULL	NULL	NULL				

13. Založení nového studenta s vložením https odkazu pomocí metody POST:

Abstract: Program uloží záznam o studentovi s vloženým https odkazem

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none"> Otevření programu Postman Otevření nového POST požadavku s URL http://108.143.193.45:8080/api/v1/students/ Vyberu záložku Body, kliknu na RAW a nastavím formát JSON Vypíši kód: <pre>{ "firstName": "https://www.google.cz/", "lastName": "odkaz", "age": 25, "email": "google@gmail.cz" }</pre> Odeslání požadavku POST Analýza status kódu Kontrola informací s databází 	<ul style="list-style-type: none"> Status 403 Forbidden

Skutečný výsledek:

Status 200 OK

```
{
  "id": 992,
  "firstName": "https://www.google.cz/",
  "lastName": "ODKAZ",
  "email": "google@gmail.cz",
  "age": 25
}
```

Result Grid					
Filter Rows:			Edit:		
	id	age	email	first_name	last_name
▶	992	25	google@gmail.cz	https://www.google...	ODKAZ
*	NULL	NULL	NULL	NULL	NULL

3. METODA DELETE:

Na základě provedených scénářů jsem objevila chyby u testů č.: 4, 5, 6 a 10 viz níže:

4. Smazání neexistujícího studenta s ID 70 pomocí metody DELETE:

Abstract: Program vypíše chybný status kód

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none"> 1. Otevření programu Postman 2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/70 3. Odeslání požadavku DELETE 4. Analýza status kódu 	<ul style="list-style-type: none"> - status kód 404 Not Found

Skutečný výsledek:

```
{
  "timestamp":
"2024-05-15T14:15:05.903+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "message": "",
  "path": "/api/v1/students/70"
}
```

5. Smazání neexistujícího studenta s ID 5 pomocí metody DELETE:

Abstract: Program vypíše chybný status kód

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/53. Odeslání požadavku DELETE4. Analýza status kódu	<ul style="list-style-type: none">- status kód 404 Not Found

Skutečný výsledek:

```
{
  "timestamp":
"2024-05-15T14:16:42.004+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "message": "",
  "path": "/api/v1/students/5"
}
```

6. Smazání studenta s ID ? pomocí metody DELETE:

Abstract: Program vypíše chybný status kód

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none">1. Otevření programu Postman2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/?3. Odeslání požadavku DELETE4. Analýza status kódu	<ul style="list-style-type: none">- status kód 400 Bad Request

Skutečný výsledek:

```
{
```

```

    "timestamp":
"2024-05-15T14:19:09.103+00:00",
    "status": 405,
    "error": "Method Not Allowed",
    "message": "",
    "path": "/api/v1/students/"
}

```

Poznámka: Platí i pro jiné znaky např. ; /

10. Smazání studenta podle % pomocí metody DELETE:

Abstract: Program vypíše chybný status kód

Kroky:	Očekávaný výsledek:
<ol style="list-style-type: none"> 1. Otevření programu Postman 2. Otevření nového DELETE požadavku s URL http://108.143.193.45:8080/api/v1/students/% 3. Odeslání požadavku DELETE 4. Analýza status kódu 	<ul style="list-style-type: none"> - status kód 400 Bad Request

Skutečný výsledek:

- tělo odpovědi nemá správný formát a obsahuje HTML stránku s nadpisem "HTTP Status 400 – Bad Request"

```

<!doctype html>
<html lang="en">

<head>
  <title>HTTP Status 400 - Bad
Request</title>
  <style type="text/css">
    body {
      font-family: Tahoma, Arial,
sans-serif;

      h1,
      h2,
      h3,

```

```
    b {
        color: white;
        background-color: #525D76;
    }

    h1 {
        font-size: 22px;
    }

    h2 {
        font-size: 16px;
    }

    h3 {
        font-size: 14px;
    }

    p {
        font-size: 12px;
    }

    a {
        color: black;
    }

    .line {
        height: 1px;
        background-color: #525D76;
        border: none;
    }
</style>
</head>

<body>
    <h1>HTTP Status 400 - Bad Request</h1>
</body>

</html>
```