

FINÁLNÍ PROJEKT

č.1



ENGETO

Autor: Pavel Švec
Datum: 2.8.2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	10
BUG REPORT	16

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	
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.

1. *abstract: positive check GET order in Postman*

open Postman

use link: http://108.143.193.45:8080/api/v1/students/268 in Postman

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"id": 268,

"firstName": "John",

"lastName": "JOSHUA",

"email": "",

"age": 34

2. *abstract: negative check GET order in Postman*

open Postman

use link: http://108.143.193.45:8080/api/v1/students/250 in Postman

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"id": 250,

"message": "This student doesn't exist."

3. *Abstract: positive check POST order in Postman*

open Postman

use link: http://108.143.193.45:8080/api/v1/students/ in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

{

"id": 1501,

```
"firstName": "Ježíšek",  
"lastName": "Svatý",  
"email": "nebe@volat.com",  
"age": 2057  
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

Status: 200 OK

```
{  
  "id": 1507,  
  "firstName": "Ježíšek",  
  "lastName": "SVATÝ",  
  "email": "nebe@volat.com",  
  "age": 2057  
}
```

ID has to be as same as DTB and it's shouldn't be as same as in this case.
Control has to be done thru DTB.

4. Abstract: duplicate check POST order in Postman

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{  
  "id": 1501,  
  "firstName": "Ježíšek",  
  "lastName": "Svatý",  
  "email": "nebe@volat.com",  
  "age": 2057  
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

```
"id": 1513,  
"message": "This student does exist."
```

ID has to be as same as DTB and it's shouldn't be as same as in this case.
Control has to be done thru DTB.

5. Abstract: negative check POST order in Postman (use space instead of letters or numbers)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": (nothing),
  "firstName": "40x space",
  "lastName": "40x space",
  "email": "40x space without @",
  "age": 10x space
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Wrong format for input of data."

6. Abstract: negative check POST order in Postman (use signs instead of letters)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": 1,
  "firstName": "+-+-+",
  "lastName": "+-+-+",
  "email": "+-+-+",
  "age": 222
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Wrong format of input data for firstName, lastName and email."

7. Abstract: negative check POST order in Postman (use letters instead of numbers for AGE)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": 1,
  "firstName": "Pavel",
  "lastName": "Novák",
  "email": "test",
  "age": fff
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Wrong format for input data for age."

8. Abstract: negative check POST order in Postman (empty ID)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": ,
  "firstName": "Pavel",
  "lastName": "Novák",
  "email": "test",
  "age": 888
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

```
{  
  "id": 1518,  
  "firstName": "Pavel",  
  "lastName": "NOVÁK",  
  "email": "test",  
  "age": 888  
}
```

9. Abstract: negative check POST order in Postman (use space instead of letters)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{  
  "id": 1,  
  "firstName": "40x space",  
  "lastName": "40x space",  
  "email": "40x space without @",  
  "age": 888  
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Too long firstName, lastName and email."

10. abstract: positive check DELETE order in Postman

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/1519> in Postman
Set DELETE and push button SEND

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"message": "Student was delete."

11. abstract: negative check DELETE order in Postman

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/1519> in Postman

Set DELETE and push button SEND

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"message": "Student can't be delete because record doesn't exist."

EXEKUCE TESTŮ

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

1. *abstract: positive check GET order in Postman*

open Postman

use link: http://108.143.193.45:8080/api/v1/students/268 in Postman

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"id": 268,

"firstName": "John",

"lastName": "JOSHUA",

"email": "",

"age": 34

result: test succesful pass

2. *abstract: negative check GET order in Postman*

open Postman

use link: http://108.143.193.45:8080/api/v1/students/250 in Postman

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"id": 250,

"message": "This student doesn't exist."

result: test failed

3. *Abstract: positive check POST order in Postman*

open Postman

use link: http://108.143.193.45:8080/api/v1/students/ in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{  
  "id": 1501,
```

```
"firstName": "Ježíšek",  
"lastName": "Svatý",  
"email": "nebe@volat.com",  
"age": 2057  
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

Status: 200 OK

```
{  
  "id": 1507,  
  "firstName": "Ježíšek",  
  "lastName": "SVATÝ",  
  "email": "nebe@volat.com",  
  "age": 2057  
}
```

ID has to be as same as DTB and it's shouldn't be as same as in this case.
Control has to be done thru DTB.

Result: test succesful pass (but ...bug report)

4. Abstract: duplicate check POST order in Postman

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{  
  "id": 1501,  
  "firstName": "Ježíšek",  
  "lastName": "Svatý",  
  "email": "nebe@volat.com",  
  "age": 2057  
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"id": 1513,

"message": "This student does exist."

ID has to be as same as DTB and it's shouldn't be as same as in this case.
Control has to be done thru DTB.

result: test failed

5. Abstract: negative check POST order in Postman (use space instead of letters or numbers)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{  
  "id": (nothing),  
  "firstName": "40x space",  
  "lastName": "40x space",  
  "email": "40x space without @",  
  "age": 10x space  
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Wrong format for input of data."

Result: test failed

6. Abstract: negative check POST order in Postman (use signs instead of letters)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{  
  "id": 1,  
  "firstName": "+-+--",  
  "lastName": "+-+--",  
  "email": "+-+--",  
}
```

```
"age": 222
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Wrong format of input data for firstName, lastName and email."

Result: test failed

7. Abstract: negative check POST order in Postman (use letters instead of numbers for AGE)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": 1,
  "firstName": "Pavel",
  "lastName": "Novák",
  "email": "test",
  "age": fff
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Wrong format for input data for age."

Result: test failed

8. Abstract: negative check POST order in Postman (empty ID)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": ,
  "firstName": "Pavel",
  "lastName": "Novák",
  "email": "test",
  "age":888
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

```
{
  "id": 1518,
  "firstName": "Pavel",
  "lastName": "NOVÁK",
  "email": "test",
  "age": 888
}
```

Result: test failed

9. Abstract: negative check POST order in Postman (use space instead of letters)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": 1,
  "firstName": "40x space",
  "lastName": "40x space",
  "email": "40x space without @",
  "age": 888
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Too long firstName, lastName and email."

Result: test failed

10. abstract: positive check DELETE order in Postman

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/1519> in Postman

Set DELETE and push button SEND

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"message": "Student was delete."

Result: test partly failed

11. abstract: negative check DELETE order in Postman

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/1519> in Postman

Set DELETE and push button SEND

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"message": "Student can't be delete because record doesn't exist."

Result: test failed

BUG REPORT

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

1. *abstract: negative check GET order in Postman*

open Postman

use link: http://108.143.193.45:8080/api/v1/students/250 in Postman

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"id": 250,

"message": "This student doesn't exist."

actual result:

status: 500 Internal Server Error

"timestamp": "2024-08-02T12:50:49.492+00:00",

"status": 500,

"error": "Internal Server Error",

"message": "",

"path": "/api/v1/students/250"

2. *Abstract: positive check POST order in Postman*

open Postman

use link: http://108.143.193.45:8080/api/v1/students/ in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{  
  "id": 1501,  
  "firstName": "Ježíšek",  
  "lastName": "Svatý",  
  "email": "nebe@volat.com",  
  "age": 2057  
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

Status: 200 OK

```
{
  "id": 1507,
  "firstName": "Ježíšek",
  "lastName": "SVATÝ",
  "email": "nebe@volat.com",
  "age": 2057
}
```

ID has to be as same as DTB and it's shouldn't be as same as in this case.

Control has to be done thru DTB.

Actual result:

Everything was OK, but on start of entering data shouldn't be necessary fulfill ID because ID is giving automaticly.

3. Abstract: duplicate check POST order in Postman

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": 1501,
  "firstName": "Ježíšek",
  "lastName": "Svatý",
  "email": "nebe@volat.com",
  "age": 2057
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"id": 1513,

"message": "This student does exist."

ID has to be as same as DTB and it's shouldn't be as same as in this case.

Control has to be done thru DTB.

Actual result:

```
{
  "id": 1513,
  "firstName": "Ježíšek",
  "lastName": "SVATÝ",
  "email": "nebe@volat.com",
  "age": 2057
}
```

ID is new (it is OK) but other data are the same as previous. Here isn't any duplicate check.

4. Abstract: negative check POST order in Postman (use space instead of letters or numbers)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": (nothing),
  "firstName": "40x space",
  "lastName": "40x space",
  "email": "40x space without @",
  "age": 10x space
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Wrong format for input data."

Actual result:

Status: 400 bad request

```
{
  "timestamp": "2024-08-09T10:48:15.275+00:00",
}
```

```

    "status": 400,
    "error": "Bad Request",
    "message": "",
    "path": "/api/v1/students/"
  }

```

5. Abstract: negative check POST order in Postman (use signs instead of letters)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```

{
  "id": 1,
  "firstName": "+-+-+",
  "lastName": "+-+-+",
  "email": "+-+-+",
  "age": 222
}

```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Wrong format of input data for firstName, lastName and email."

Actual result:

```

{
  "id": 1514,
  "firstName": "+-+-+",
  "lastName": "+-+-+",
  "email": "+-+-+",
  "age": 222
}

```

Here ins't control of quality of entering data. Both names have to be fulfill with letters, not signs. Email has to be fulfill with one @, not without it.

6. Abstract: negative check POST order in Postman (use letters instead of numbers for AGE)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": 1,
  "firstName": "Pavel",
  "lastName": "Novák",
  "email": "test",
  "age": fff
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Wrong format for input data for age."

Actual result:

```
{
  "timestamp": "2024-08-09T11:09:26.796+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "",
  "path": "/api/v1/students/"
}
```

7. Abstract: negative check POST order in Postman (empty ID)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{
  "id": ,
  "firstName": "Pavel",

```

```
"lastName": "Novák",  
"email": "test",  
"age": 888  
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

```
{  
  "id": 1518,  
  "firstName": "Pavel",  
  "lastName": "NOVÁK",  
  "email": "test",  
  "age": 888  
}
```

Actual result:

```
{  
  "timestamp": "2024-08-09T11:16:22.477+00:00",  
  "status": 400,  
  "error": "Bad Request",  
  "message": "",  
  "path": "/api/v1/students/"  
}
```

8. Abstract: negative check POST order in Postman (use space instead of letters)

open Postman

use link: <http://108.143.193.45:8080/api/v1/students/> in Postman

Click Body, next click raw and choose JSON language on same line.

Next enter this:

```
{  
  "id": 1,  
  "firstName": "40x space",
```

```
"lastName": "40x space",
"email": "40x space without @",
"age": 888
}
```

to the editing place and push SEND button.

expected result:

Postman will work without issue.

used link and fulfilling BODY should show this results:

status: 200 OK

"message": "Too long firstName, lastName and email."

Actual result:

```
{
  "id": 1519,
  "firstName": " ",
  "lastName": " ",
  "email": " ",
  "age": 888
}
```

9. abstract: positive check DELETE order in Postman
open Postman
use link: <http://108.143.193.45:8080/api/v1/students/1519> in Postman
Set DELETE and push button SEND

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"message": "Student was delete."

Actual result:

Status: 200 OK

Without any message. Record was deleted from DTB.

10. abstract: negative check DELETE order in Postman
open Postman
use link: <http://108.143.193.45:8080/api/v1/students/1519> in Postman
Set DELETE and push button SEND

expected result:

Postman will work without issue.

used link should show this results:

status: 200 OK

"message": "Student can't be deleted because record doesn't exist."

Actual result:

Status: 500 Internal Server error

```
{  
  "timestamp": "2024-08-09T12:00:51.602+00:00",  
  "status": 500,  
  "error": "Internal Server Error",  
  "message": "",  
  "path": "/api/v1/students/1519"  
}
```

Mam to uznać