FINÁLNÍ PROJEKT   
 č.1



Autor:Václav Klecan

Datum:28.9.2024

OBSAH

[**ZADÁNÍ**](#_heading=h.30j0zll) **3**

[**TESTOVACÍ SCÉNÁŘE**](#_heading=h.2et92p0) **4**

[**EXEKUCE TESTŮ**](#_heading=h.tyjcwt) **5**

[**BUG REPORT**](#_heading=h.3dy6vkm) **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: sem7fngq10dy6oj7iz6x  Password: pscale\_pw\_C8LU0FVofFjcbzuVga2lgCHE9RrI0VDL2zI2GpWfsnn |
| --- | --- |
| REST-API | http://108.143.193.45:8080/api/v1/students/ |

## Poznámky:

Nezapomeňte, že v IT se data musí někde uložit a poté získat. Proto ověřte, že data jsou správně uložena a získávána z databáze.

Nezapomeňte do testovacích scénářů uvést testovací data, očekávaný výsledek včetně těla odpovědi a stavových kódů.

:

# TESTOVACÍ SCÉNÁŘE

*Na základě uvedených testovacích scénářů jsem ověřil(a) funkčnost aplikace.*

*1)do Postman pošlu GET dotaz na získání všech studentů a překontroluju výsledky se seznamem studentů*

*2)přidáme si studenta a po přidání si skrz mysql workbench překontrolujeme jestli sedí údaje*

*3)smažeme studenta kterého jsme zakládali v minulém bodě a zkontrolujeme jestli byl opravdu smazán*

*4)vytvoříme si nového studenta ,ale do parametru age zadáme nějaký string a zkontrolujeme výsledek*

*5)novému studentovi z bodu 4 změníme parametry firstName a Age a zkontrolovat výsledek*

*6)do postman pošlu GET dotaz na studenta který neexistuje*

*7)vytvoříme studenta ,ale nezadáme mu věk a zkontrolujeme výsledek*

*8)vytvoříme si studenta ,ale do parametru firstname zadáme nějaké číslo*

*9)smažeme studenta který neexistuje*

*10)zkusit smazat studenty bez ID a překontrolovat výstup ,nemělo by se nic smazat*

*11)v postman metodou GET zadáme na konec URL nějaké písmenko a zkontrolujeme výstup*

*12)metodou POST vytvoříme studenta a do parametru age vložíme 2000*

*13)vytvoříme si více studentů s identickými údaji (duplikace jména není problém ,ale email by neměl být stejný)*

EXEKUCE TESTŮ

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

*1)Pomocí Postman jsem poslal GET dotaz na adresu* [*http://108.143.193.45:8080/api/v1/students/*](http://108.143.193.45:8080/api/v1/students/) *a překontroloval jsem prvních 5 lidí že informace souhlasí zasláním tohoto dotazu SELECT \* FROM student na sql Workbench*

*2)Pomocí POST metody jsem si v Postman přidal studenta*

*{*

*"firstName": "experiment",*

*"lastName": "socialni",*

*"email": "experiment@gmail.com",*

*"age": 10*

*}*

*poté jsem dostal status kod 200 a vygenerované id (id = 1852)*

* *{*
* *"id": 1852,*
* *"firstName": "experiment",*
* *"lastName": "SOCIALNI",*
* *"email": "experiment@gmail.com",*
* *"age": 10*
* *}*

*a po přidání jsem zpětně zkontroloval jestli byl vytvořen skrz mysql Workbench dotazem SELECT \* FROM student WHERE id = 1852*

*chyba: lastName bylo zadáno jako socialni ale po přidání se změnilo na SOCIALNI*

*3)Metodou DELETE jsem z Postman odstranil studenta s id = 1852 ,do mysql Workbench jsem poslal žádost SELECT \* FROM student WHERE id = 1852*

*která nenavrátila nic ,dle očekávání.*

# 4)V Postman jsem si vytvořil šablonu nového studenta kde jsem místo čísla zadal do parametru age string

{

"firstName": "monstr",

"lastName": "trak",

"email": "monstrik@gmail.com",

"age": "rrrrr"

}

pomocí POST jsem ho vytvořil ,byl navrácen status 400 ,error = bad request

5)vytvořenému studentovi z bodu 4 ,jsem změnil firstName na velkej a Age na 15 pomocí POST metody a změny byly provedeny úspěšně

6)do postmana jsem poslal GET dotaz <http://108.143.193.45:8080/api/v1/students/15225816155> ,postman vrátil {

"timestamp": "2024-10-12T10:36:40.292+00:00",

"status": 500,

"error": "Internal Server Error",

"message": "",

"path": "/api/v1/students/15225816155"

}

7)v postman jsem metodou POST poslal žádost o vytvoření studenta

{

"firstName": "misa",

"lastName": "rez",

"email": "misarez@gmail.com",

"age":

}

postman vrátil

{

"timestamp": "2024-10-12T10:45:26.571+00:00",

"status": 400,

"error": "Bad Request",

"message": "",

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

}

8)do postman jsem poslal POST dotaz na vytvoření studenta

{

"firstName": 8,

"lastName": "doe",

"email": "jodoe@gmail.com",

"age": 12

}

a postman vrátil

{

"id": 1869,

"firstName": "8",

"lastName": "DOE",

"email": "jodoe@gmail.com",

"age": 12

}

9) pomocí DELETE metody smažeme studenta s ID 8911919655558 (student s tímto id neexistuje)

postman nám vráti odpověď

{

"timestamp": "2024-10-12T10:56:42.753+00:00",

"status": 500,

"error": "Internal Server Error",

"message": "",

"path": "/api/v1/students/8911919655558"

}

10) metodou DELETE pošleme dotaz na URL <http://108.143.193.45:8080/api/v1/students/> ,postman nám vrátí {

"timestamp": "2024-10-12T11:47:17.682+00:00",

"status": 405,

"error": "Method Not Allowed",

"message": "",

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

}

11)vybereme si metodu GET a na konec URL zadáme nějaké libovolné písmenko ,postman nám vrátí

{

"timestamp": "2024-10-12T11:11:25.473+00:00",

"status": 400,

"error": "Bad Request",

"message": "",

"path": "/api/v1/students/a"

}

12)pomocí metody POST zadáme studentovi věk 2000 ,postman nám vrátí že student byl úspěšně vytvořen

13)*pomocí POST metody si vytvoříme více studentů s identickými údaji*

*{*

*"firstName": "mr",*

*"lastName": "duplikat",*

*"email": "stejnej@gmail.com",*

*"age": 7*

*}*

*postman nám vždy vytvoří nového studenta s jiným ID ,ale s kompletně identickými údaji*

# BUG REPORT

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

*1)POST metoda způsobuje že parametr lastName bude vždy v uppercase*

*2) Když si metodou GET vyčteme studenta který neexistuje tak nám postman vrátí*

"status": 500,

"error": "Internal Server Error",

,ale měl by nám vrátit status 404 not found značící že student neexistuje

3)pokud do parametru age zadáme integer tak se přemění ve string po použítí metody POST

4)pokud smažeme studenta který neexitsuje postman nám vrátí

"status": 500,

"error": "Internal Server Error",

,ale měl by navracet status: 404 not found

5)při tvoření studenta není kontrolován vstup na věk ,můžeme vytvořit studenta který je 2000 let starý

6)můžeme si vytvořit libovolný počet studentů s identickými údaji (kritický je email) ,jediné co se bude lišit je ID