

FINÁLNÍ PROJEKT

č.1



ENGETO

Autor: Nikola Pokorná
Datum: 3.11.2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	11
BUG REPORT	12

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: im3s0o2b4w0lpcboxk0k Password: pscale_pw_8Nc3bfP1t3ECEJm8cGPze47M egBUWQzgH7J8XsLghBi
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. GET_01

Nadpis: Načtení všech záznamů

Verze: 1.0

Kroky reprodukce:

- a) Otevřela jsem aplikaci Postman.
- b) Vybrala jsem metodu GET.
- c) Zadala jsem do vyhledávacího řádku
"http://108.143.193.45:8080/api/v1/students".
- d) Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 200 (OK)
- obsah odpovědi: JSON pole s daty v body (minimálně jeden záznam)

Aktuální výsledek:

- stavový kód: 200 (OK)
- obsah odpovědi: JSON pole s daty v body

2. GET_02

Nadpis: Načtení určitého existujícího studenta podle existujícího ID

Verze: 1.0

Kroky reprodukce:

- a) Otevřela jsem aplikaci MySQL Workbench a následně jsem otevřela databázi se zadanými údaji.
- b) Zadala jsem do příkazového řádku "SELECT * FROM student LIMIT 100; " a vybrala si náhodného existujícího studenta s ID 351.
- c) Otevřela jsem aplikaci Postman.
- d) Vybrala jsem metodu GET.
- e) Zadala jsem vyhledávacího řádku
"http://108.143.193.45:8080/api/v1/students/351".
- f) Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 200 (OK)
- obsah odpovědi: JSON pole studenta s ID 351 a všemi stejnými údaji jako v MySQL Workbench.

Aktuální výsledek:

- stavový kód: 200 (OK)

- obsah odpovědi: JSON pole studenta s ID 351 a všemi stejnými údaji jako v MySQL Workbench.

3. GET_03

Nadpis: Načtení studenta podle neexistujícího ID - neexistující celé číslo

Verze: 1.0

Kroky reprodukce:

- a) Otevřela jsem aplikaci MySQL workbench a následně jsem otevřela databázi se zadanými údaji.
- b) Zadala jsem do příkazového řádku "SELECT * FROM student LIMIT 100; " a vybrala si neexistujícího studenta s ID 355.
- c) Otevřela jsem aplikaci Postman.
- d) Vybrala jsem metodu GET.
- e) Zadala jsem vyhledávacího řádku "http://108.143.193.45:8080/api/v1/students/355".
- f) Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 404 (Not Found)
- obsah odpovědi: nenalezení žádného JSON pole studenta s ID 355.

Aktuální výsledek:

- stavový kód: 500 (Internal Server Error)
- obsah odpovědi: nenalezení žádného JSON pole studenta s ID 355.

4. GET_04

Nadpis: Načtení studenta podle neexistujícího ID - neexistující desetinné číslo

Verze: 1.0

Kroky reprodukce:

- a) Otevřela jsem aplikaci Postman.
- b) Vybrala jsem metodu GET.
- c) Zadala jsem vyhledávacího řádku "http://108.143.193.45:8080/api/v1/students/355.5".
- d) Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 400 (Bad request)
- obsah odpovědi: nenalezení žádného JSON pole studenta s ID 355,5.

Aktuální výsledek:

- stavový kód: 400 (Bad request)
- obsah odpovědi: nenalezení žádného JSON pole studenta s ID 355,5.

5. GET_05

Nadpis: Načtení studenta podle neexistujícího ID - speciální znaky

Verze: 1.0

Kroky reprodukce:

- a) Otevřela jsem aplikaci Postman.
- b) Vybrala jsem metodu GET.
- c) Zadala jsem vyhledávacího řádku
"http://108.143.193.45:8080/api/v1/students/3*3".
- d) Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 400 (Bad request)
- obsah odpovědi: nenalezení žádného JSON pole studenta s ID 3*3.

Aktuální výsledek:

- stavový kód: 400 (Bad request)
- obsah odpovědi: nenalezení žádného JSON pole studenta s ID 3*3.

6. POST_01

Nadpis: Zadáání všech parametrů ve správném formátu - first_name, last_name, email, age

Verze: 1.0

Kroky reprodukce:

- a) Otevřela jsem aplikaci Postman.
- b) Vybrala jsem metodu POST.
- c) Klikla jsem na "body".
- d) Klikla jsem na "raw".
- e) Vybrala jsem formát JSON.
- f) Vložila jsem do body:

```
{
  "firstName": "Nikola",
  "lastName": "Pokorna",
  "email": "nikolapokorna@centrum.cz",
  "age": 25
}
```
- g) Klikla jsem na odeslat.
- h) Podle přiděleného ID (2090) jsem zkontrolovala v aplikaci MySQL workbench, že se student opravdu vytvořil a uložil do databáze (Zadala jsem příkaz "SELECT * FROM student where id=2090;").

Očekávaný výsledek:

- stavový kód: 200 (OK)
- obsah odpovědi: Vytvořil se student s daty zadanými výše.

Aktuální výsledek:

- stavový kód: 200 (OK)
- obsah odpovědi: Vytvořil se student s daty zadanými výše.

7. POST_02

Nadpis: Zadáání pouze jednoho parametru

Verze: 1.0

Kroky reprodukce:

- a) Otevřela jsem aplikaci Postman.
- b) Vybrala jsem metodu POST.
- c) Klikla jsem na "body".
- d) Klikla jsem na "raw".
- e) Vybrala jsem formát JSON.
- f) Vložila jsem do body:

```
{  
  "firstName": "Nikola",  
  "lastName":  
  "email":  
  "age":  
}
```

- g) Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 400 (Bad Request)
- obsah odpovědi: Student se nevytvoří.

Aktuální výsledek:

- stavový kód: 400 (Bad Request)
- obsah odpovědi: Student se nevytvoří.

8. POST_03

Nadpis: Zadání stringu do age

Verze: 1.0

Kroky reprodukce:

- a) Otevřela jsem aplikaci Postman.
- b) Vybrala jsem metodu POST.
- c) Klikla jsem na "body".
- d) Klikla jsem na "raw".
- e) Vybrala jsem formát JSON.
- f) Vložila jsem do body:

```
{  
  "firstName": "Nikola",  
  "lastName": "Pokorna",  
  "email": "nikolapokorna@centrum.cz",  
  "age": "dvacetpet"  
}
```

- g) Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 400 (Bad Request)
- obsah odpovědi: Student se nevytvoří.

Aktuální výsledek:

- stavový kód: 400 (Bad Request)

- obsah odpovědi: Student se nevytvoří.

9. POST_04

Nadpis: Zadání desetinného čísla do age

Verze: 1.0

Kroky reprodukce:

- Otevřela jsem aplikaci Postman.
- Vybrala jsem metodu POST.
- Klikla jsem na "body".
- Klikla jsem na "raw".
- Vybrala jsem formát JSON.
- Vložila jsem do body:

```
{  
  "firstName": "Nikola",  
  "lastName": "Pokorna",  
  "email": "nikolapokorna@centrum.cz",  
  "age": 25,5  
}
```

- Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 400 (Bad Request)
- obsah odpovědi: Student se nevytvoří.

Aktuální výsledek:

- stavový kód: 400 (Bad Request)
- obsah odpovědi: Student se nevytvoří.

10. POST_05

Nadpis: Zadání čísla do first_name, last_name a email

Verze: 1.0

Kroky reprodukce:

- Otevřela jsem aplikaci Postman.
- Vybrala jsem metodu POST.
- Klikla jsem na "body".
- Klikla jsem na "raw".
- Vybrala jsem formát JSON.
- Vložila jsem do body:

```
{  
  "firstName": 1,  
  "lastName": 5,  
  "email": 5,  
  "age": 25  
}
```

- Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 400 (Bad Request)
- obsah odpovědi: Student se nevytvoří.

Aktuální výsledek:

- stavový kód: 200 (OK)
- obsah odpovědi: Student se vytvoří.

11. POST_06

Nadpis: Zadání emailu v nesprávném formátu (bez zavináče, bez internetové domény nebo bez části před @)

Verze: 1.0

Kroky reprodukce:

- Otevřela jsem aplikaci Postman.
- Vybrala jsem metodu POST.
- Klikla jsem na "body".
- Klikla jsem na "raw".
- Vybrala jsem formát JSON.
- Vložila jsem do body:

```
{
  "firstName": "Nikola",
  "lastName": "Pokorna",
  "email": "@centrum.cz",
  "age": 25
}
```

(popř. další hodnoty: "nikolapokorna@" ; "nikolapokornacentrum.cz")

- Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 400 (Bad Request)
- obsah odpovědi: Student se nevytvoří.

Aktuální výsledek:

- stavový kód: 400 (Bad Request)
- obsah odpovědi: Student se nevytvoří.

12. DELETE_01

Nadpis: Smazání existujícího studenta s platným ID

Verze: 1.0

Kroky reprodukce:

- Otevřela jsem aplikaci Postman.
- Vybrala jsem metodu DELETE.
- Zadala jsem vyhledávacího řádku
"http://108.143.193.45:8080/api/v1/students/2090".
- Klikla jsem na odeslat.

- e) Zkontrolovala jsem ještě v aplikaci My SQL workbench, že se opravdu smazal.

Očekávaný výsledek:

- stavový kód: 200 (OK)
- očekávaná odpověď: žádná zobrazená data

Aktuální výsledek:

- stavový kód: 200 (OK)
- očekávaná odpověď: žádná zobrazená data

13. DELETE_02

Nadpis: Smazání neexistujícího studenta s neplatným ID

Verze: 1.0

Kroky reprodukce:

- a) Otevřela jsem aplikaci Postman.
- b) Vybrala jsem metodu DELETE.
- c) Zadala jsem vyhledávacího řádku
"http://108.143.193.45:8080/api/v1/students/355" (neexistující student).
- d) Klikla jsem na odeslat.

Očekávaný výsledek:

- stavový kód: 404 (Not Found)
- očekávaná odpověď: žádná zobrazená data

Aktuální výsledek:

- stavový kód: 500 (Internal Server Error)
- očekávaná odpověď: žádná zobrazená data

EXEKUCE TESTŮ

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

ID scénáře	METODA	ENDPOINT	OČEKÁVANÝ VÝSLEDEK	SKUTEČNÝ VÝSLEDEK	STATUS
GET_01	GET	/students	200 OK, seznam studentů	200 OK, seznam studentů	Passed
GET_02	GET	/students/351	200 OK, student ID 351	200 OK, student ID 351	Passed
GET_03	GET	/students/355	404 Not Found, žádná data	500 Internal Server Error, žádná data	Failed
GET_04	GET	/students/355.5	400 Bad Request, žádná data	400 Bad Request, žádná data	Passed
GET_05	GET	/students/3*3	400 Bad Request, žádná data	400 Bad Request, žádná data	Passed
POST_01	POST	/students	200 OK, student ID 2090	200 OK, student ID 2090	Passed
POST_02	POST	/students	400 Bad Request, žádná data	400 Bad Request, žádná data	Passed
POST_03	POST	/students	400 Bad Request, žádná data	400 Bad Request, žádná data	Passed
POST_04	POST	/students	400 Bad Request, žádná data	400 Bad Request, žádná data	Passed
POST_05	POST	/students	400 Bad Request, žádná data	200 OK, student ID 2191	Failed

POST_06	POST	/students	400 Bad Request, žádná data	400 Bad Request, žádná data	Passed
DELETE_01	DELETE	/students/2090	200 OK, žádná data	200 OK, žádná data	Passed
DELETE_02	DELETE	/students/355	404 Not Found, žádná data	500 Internal Server Error, žádná data	Failed

BUG REPORT

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

BUG ID: BUG_01

Popis: GET_03 požadavek na neexistující záznam vrací chybu 500 místo 404.

Kroky k reprodukci:

1. Odeslat GET požadavek na **/students/355** (záznam s ID 355 neexistuje).
2. Zkontrolovat odpověď a stavový kód.

Očekávaný výsledek: API by mělo vrátit stavový kód 404 (Not Found), protože záznam neexistuje.

Skutečný výsledek: API vrací kód 500 (Internal Server Error).

Priorita: Střední (tato chyba ovlivňuje správnou informovanost uživatele při neexistujícím záznamu).

BUG ID: BUG_02

Popis: POST_05 Zadání čísla do polí first_name, last_name a email umožňuje vytvoření studenta místo vrácení chyby.

Kroky k reprodukci:

1. Odeslat metodou POST v body následující JSON data: { "firstName": 1, "lastName": 5, "email": 5, "age": 25 }.
2. Zkontrolovat stavový kód a zajistit, aby se nedala odeslat pouze čísla ve first_name, last_name a email.

Očekávaný výsledek: API by mělo vrátit stavový kód 400 (Bad request), protože student nemůže mít jméno a email pouze jako číslíci.

Skutečný výsledek: API vrací kód 200 (OK) a vytvoří studenta.

Priorita: Vysoká (chyba umožňuje neplatná data, což může vést k nekonzistenci v databázi a nesprávným údajům o studentech).

BUG ID: BUG_03

Popis: DELETE_02 smazání neexistujícího studenta s neplatným ID vrací chybu 500 místo 404.

Kroky k reprodukci:

3. Odeslat DELETE požadavek na `/students/355` (záznam s ID 355 neexistuje).
4. Zkontrolovat odpověď a stavový kód.

Očekávaný výsledek: API by mělo vrátit stavový kód 404 (Not Found), protože záznam neexistuje.

Skutečný výsledek: API vrací kód 500 (Internal Server Error).

Priorita: Střední (tato chyba ovlivňuje správnou informovanost uživatele při neexistujícím záznamu).