

FINÁLNÍ PROJEKT

č.1



ENGETO

Autor Andrea Tužáková:
Datum:7.12.2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	8
BUG REPORT	10

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	Default scheme: qa_demo Host: aws.connect.psdb.cloud Port: 3306
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

GET

1. T: Správne fungovanie prepojenia medzi databázou a programom Postman

- P:
1. Zadáš do Postmana link GET <http://108.143.193.45:8080/api/v1/students/>
 2. Send
 3. Zkontroluj údaje s údajmi v databáze

Očakávaný výsledok: status 200 ok

Zobrazenie údajov v body zhodných s databázou

2. T: Overenie prítomnosti študenta v zadanej databáze

- P:
1. Vyber si z databáze študenta s prideleným ID
 2. Zadáš do Postmana GET <http://108.143.193.45:8080/api/v1/students/<id-studenta>>
 3. Send
 4. Zkontroluj údaje s údajmi v databáze

Očakávaný výsledok: status 200 OK

Údaje v body sa zhodujú s databázou.

```
{  
  "id": 363,  
  "firstName": "John",  
  "lastName": "DO",  
  "email": "john.doe@gmail.com",  
  "age": 34  
}
```

3. T: Overenie prítomnosti neexistujúceho študenta v zadanej databáze

- P:
1. Pozri sa v databáze študentov na ID, a vyber si také ktoré tam nieje
 2. Zadáš do postmana GET <http://108.143.193.45:8080/api/v1/students/<id-studenta>>
 3. Send
 4. Skontroluj že sa v databáze naozaj nenachádza študent s daným ID

Očakávaný výsledok: status 404 – not found

Prázdné body, hláška – Študent s ID neexistuje/není v databáze.

DELETE

4. T: Zmazanie existujúceho študenta

- P:
1. Vytvor si v Postmanovi pomocou funkcie Post nového študenta
 2. Send
 3. Otvor databázu a nájdi študenta s novým ID
 4. Zmen z Post na Delete a pridaj link
<http://108.143.193.45:8080/api/v1/students/<id-studenta>>(id študenta ktorý bol vytvorený)
 5. Send
 6. Skontroluj v databáze

Očakávaný výsledok: status 200 OK

Prázdne body.

5. T: Zmazanie neexistujúceho študenta

- P:
1. Otvor databázu a pozri sa ktoré ID sa tam nenachádzajú
 2. Otvor Postmana, zadaj Delete a pridaj link
<http://108.143.193.45:8080/api/v1/students/<id-studenta>>(id študenta ktoré se v databáze nenachádza)
 3. Send

Očakávaný výsledok: status 404 – not found

Prázdne body, hláška – Student s ID sa nevymazal pretože neexistuje.

POST

6. T: Vytvorenie nového študenta zadaním správnych údajov

- P:
1. Otvoríš program Postman-metóda POST-body-zaškrtnieš raw-a vyberieš JSON
 2. Vytvorí sa ti šablóna a doplníš správne údaje-firstname,lastname,email,age
 3. Send
 4. V body sa ti zobrazí vytvorený študent
 5. Kontrola v databáze

Očakávaný výsledok: status 200 OK

Študent sa zobrazí v databáze aj v body Postmana.

7. T: Vytvorenie nového študenta odstránením jedného celého parametru

- P:
1. Otvoríš program Postman-metóda POST-body-zaškrtnieš raw-a vyberieš JSON
 2. Vytvorí sa ti šablóna a doplníš správne údaje-firstname,lastname,email
 3. Vymažeš celý riadok s "age"
 4. Send

Očakávaný výsledok: status 400 Bad request

```
{  
  "timestamp": "2024-12-02T10:58:18.761+00:00",  
  "status": 400,  
  "error": "Bad Request",  
  "message": "",  
  "path": "/api/v1/students/"  
}
```

8. T: Vytvorenie študenta ak nevyplníš parameter lastname

- P:
1. Otvoríš program Postman-metóda POST-body-zaškrtnieš raw-a vyberieš JSON
 2. Vytvorí sa ti šablóna a doplníš správne údaje - firstname, email, age
 3. Necháš prázdny údaj lastname
 4. Send

Očakávaný výsledok: status 400 Bad request

9. T: Vytvorenie firstName a lastName študenta pridaním špeciálnych znakov a čísel

- P:
1. Otvoríš program Postman-metóda POST-body-zaškrtnieš raw-a vyberieš JSON
 2. Vytvorí sa ti šablóna a doplníš správne údaj - email, age
 3. Nahradíš firstName a lastName číslami a znakmi



```
{  
  "firstName": "Ž1+n-ta",  
  "lastName": "D1-.@lha",  
  "email": "zaneta@mail.com",  
  "age": -1  
}
```

4. Send
5. Kontrola v databáze

Očakávaný výsledok: status 404 bad request

Hláška - Neplatné znaky v parametroch firstName a lastName

10. T: Nesprávny formát emailu

- P:
1. Otvoríš program Postman-metóda POST-body-zaškrtnieš raw-a vyberieš JSON

2. Vytvorí sa ti šablóna a doplniť správne údaje-firstname ,last name, age
3. Zadáš nesprávny formát emailu
4. Send
5. Kontrola v databáze

● none ● form-data ● x-www-form-urlencoded ● **raw**

```
1  {
2    "firstName": "Žaneta",
3    "lastName": "Dlha",
4    "email": "zanetamail.com",
5    "age": 18
6  }
```

Očakávaný výsledok: 404 bad request

Hláška - Neplatná forma emailovej adresy

11. T: Nereálny údaj o veku

- P:
- 1.Otvoríš program Postman-metóda POST-body-zaškrtnieš raw-a vyberieš JSON
 2. Vytvorí sa ti šablóna a doplniť správne údaje-firstname, last name, email
 3. Zadáš nereálny vek
 4. Send
 5. Kontrola v databáze

```
1  {
2    "firstName": "Žaneta",
3    "lastName": "Dlha",
4    "email": "zaneta@mail.com",
5    "age": -1
6  }
```

Očakávaný výsledok: 404 bad request

Hláška - Nesprávny údaj o veku.

EXEKUCE TESTŮ

1. Reálný výsledek: Status 200 Ok

Zobrazení údajů v body Postmana zhodných s databází

2. Reálný výsledek: status 200 OK

Údaje v body se shodují s databází

3. Reálný výsledek: status 500-Internal Server Error

body:

```
{  
  "timestamp": "2024-11-28T10:41:17.052+00:00",  
  "status": 500,  
  "error": "Internal Server Error",  
  "message": "",  
  "path": "/api/v1/students/12%0A"  
}
```

4. Reálný výsledek: status 200 OK, prázdné body.

5. Reálný výsledek: status 500 Internal server error

```
{  
  "timestamp": "2024-11-28T11:46:53.939+00:00",  
  "status": 500,  
  "error": "Internal Server Error",  
  "message": "",  
  "path": "/api/v1/students/12"  
}
```

6. Reálný výsledek: status 200 OK

Študent sa zobrazí v databáze aj v body Postmana.

7. Reálný výsledek-status 500 Internal Server Error

```
{  
  "timestamp": "2024-12-02T10:59:28.200+00:00",  
  "status": 500,  
  "error": "Internal Server Error",  
  "message": "",  
  "path": "/api/v1/students/"  
}
```

8. Reálný výsledek: status: 400 Bad request

9. Reálny výsledok: status 200 ok

V body aj v databáze sa zapísal študent so zlým menom a priezviskom

10. Reálny výsledok: status 200 ok

V body aj v databáze sa zapísal študent s nespravným tvarom emailu

11. Reálny výsledok: status 200 ok

V body aj v databáze sa zapísal študent s nereálnym vekom

BUG REPORT

Na základě provedených scénářů jsem objevila uvedené chyby aplikace.

bug	probability	severity	test case number	mitigace
Pri overení neexistujúceho študenta sa objaví status 500	vysoká	3	3	Chybu odstráni programátor Pred zadáním si overiť že daný študent je naozaj v databáze. Je potrebné nastaviť hlášku o tom že študent neexistuje a status bol 200
Pri vymazaní neexistujúceho študenta sa objaví status 500	vysoka	2	5	Chybu odstráni vývojár. Je potrebné nastaviť hlášku o tom že študent neexistuje a status bol 200
Pri vymazaní parametru u študenta sa objaví status 500	vysoká	3	7	Chybu odstráni vývojár.
Žiadna reakcia programu na neplatné znaky v mene a priezvisku-status 200	vysoká	3	9	Chybu odstráni vývojár. Zakázať čísla a neplatné znaky v mene a priezvisku
Žiadna reakcia na nesprávny formát emailovej adresy status 200	vysoká	2	10	Chybu odstráni vývojár. Upraviť kód tak, aby prijal len správny formát emailovej adresy a nastaviť upozornenia v prípade že email bude v zlom formáte
Žiadna reakcia pri udaní nereálneho veku-status 200	vysoká	3	11	Chybu odstráni vývojár. Je potrebné nastaviť vekové rozpätie