

# FINÁLNÍ PROJEKT

## č.1



Datum: 8.8.2024

## OBSAH

<b>ZADÁNÍ</b>	<b>3</b>
<b>TESTOVACÍ SCÉNÁŘE</b>	<b>4</b>
<b>EXEKUCE TESTŮ</b>	<b>5</b>
<b>BUG REPORT</b>	<b>5</b>

# 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 ověřila funkčnost aplikace.*

## **Abstarct: Získání všech studentů pomocí metody GET**

Krok 1.) Otevři Postmana

Krok 2.) Zvol metodu GET a vlož do pole <http://108.143.193.45:8080/api/v1/students>

Krok 3.) Stiskni tlačítko Send

Očekávaný výsledek:

- Aplikace se otevře a vypadá v pořádku
- Po vložení <http://108.143.193.45:8080/api/v1/students> je možné stisknout tlačítko Send
- Vypíše se celá databáze studentů

## **Abstarct: Získání studenta pomocí jeho ID**

Krok 1.) Otevři Postmana

Krok 2.) Zvol metodu GET a vlož do pole <http://108.143.193.45:8080/api/v1/students/584>

Krok 3.) Stiskni tlačítko Send

Očekávaný výsledek:

- Aplikace se otevře a vypadá v pořádku
- Po vložení <http://108.143.193.45:8080/api/v1/students/584> je možné stisknout tlačítko Send
- Vypíše se student se zvoleným ID 584

## **Abstarct: Získání studenta, který neexistuje**

Krok 1.) Otevři Postmana

Krok 2.) Zvol metodu GET a vlož do pole ID, které není v databázi např. 1  
<http://108.143.193.45:8080/api/v1/students/1>

Krok 3.) Stiskni tlačítko Send

Očekávaný výsledek:

- Aplikace se otevře a vypadá v pořádku
- Po vložení <http://108.143.193.45:8080/api/v1/students/1> je možné stisknout tlačítko Send
- Vypíše se chyba **500Internal Server Error**

## **Abstarct: Vytvoření nového studenta s platnými údaji**

Krok 1.) Otevři Postmana

Krok 2.) Zvol metodu POST a vlož do pole <http://108.143.193.45:8080/api/v1/students>

Krok 3.) Stiskni tlačítko Send

Krok 4.) Přejdi do sekce "Body"

Krok 5.) vlož šablonu a vyplň všechny potřebné údaje

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

Krok 6.) Stiskni tlačítko send

Očekávaný výsledek:

- Aplikace se otevře a vypadá v pořádku
- Po vložení <http://108.143.193.45:8080/api/v1/students> je možné stisknout tlačítko Send
- Po vyplnění údajů se uloží nov student do databáze

### **Abstarct: Vytvoření studenta s chybějícími povinnými údaji**

Krok 1.) Otevři Postmana

Krok 2.) Zvol metodu POST a vlož do pole <http://108.143.193.45:8080/api/v1/students>

Krok 3.) Stiskni tlačítko Send

Krok 4.) Přejdi do sekce body a vlož

```
{  
  "id": ,  
  "firstName": "",  
  "age":  
}
```

Krok 5.) Vyplň údaje o studentovi a vynech "last name" a "email" a stiskni tlačítko send

Očekávaný výsledek:

- Aplikace se otevře a vypadá v pořádku
- Po vložení <http://108.143.193.45:8080/api/v1/students> je možné stisknout SEND
- Po vyplnění údajů v sekci Body a po odeslání, se ukáže chyba "ERROR"

### **Abstarct: Vytvoření studenta s duplicitním emailem**

Krok 1.) Otevři Postmana

Krok 2.) Zvol metodu POST a vlož do pole <http://108.143.193.45:8080/api/v1/students>

Krok 3.) Stiskni tlačítko Send

Krok 4.) Přejdi do sekce "Body"

Krok 5.) Vlož šablonu a vyplň všechny potřebné údaje, do pole email vlož už existující email v databázi

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

Krok 6.) Stiskni tlačítko send

Očekávaný výsledek:

- Aplikace se otevře a vypadá v pořádku
- Po vložení <http://108.143.193.45:8080/api/v1/students> je možné stisknout tlačítko Send
- Po vyplnění šablony, duplicitního emailu a stisknutí tlačítka 'SEND', se vytvoří chyba "ERROR"

### **Abstarct: Odstranění existujícího studenta**

Krok 1.) Otevřu si postmana

Krok 2.) Zadáám <http://108.143.193.45:8080/api/v1/students> Zvolím metodu 'DELETE' a vložím si Id existujícího studenta: např.

Krok 3.) Stisknu tlačítko SEND

Očekávaný výsledek:

- Aplikace se otevře a vypadá v pořádku

-Po stisknutí tlačítka send se existující student vymaže z databáze

### **Abstarct: Odstranění nexistujícího studenta**

Krok 1.) Otevřu si postmana

Krok 2.) Zadáám <http://108.143.193.45:8080/api/v1/students> Zvolím metodu 'DELETE' a vložím si Id neexistujícího studenta.

Krok 3.) Stisknu tlačítko SEND

Očekávaný výsledek:

- Aplikace se otevře a vypadá v pořádku

-Po stisknutí tlačítka send se ukáže chyba, protože takový student neexistuje



# BUG REPORT

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

**Abstract:** Vytvoření studenta s vybraným id

- 1) Otevřu Postmana a zvolím metodu POST
- 2) Přejdu do sekce body a zadám:

```
{  
  "id":1,  
  "firstName": "josef",  
  "lastName": "pacička",  
  "email": "josejosef@josef.com",  
  "age":18  
}
```

- 3) Stisknu Post

Očekávaný výsledek: Student se založí pod vybraným ID

Skutečný výsledek: Student se uložil s jiným náhodným ID

```
{  
  "id": 1789,  
  "firstName": "josef",  
  "lastName": "PACIČKA",  
  "email": "josejosef@josef.com",  
  "age": 18  
}
```

**Abstract:** Vytvoření studenta s duplicitním emailem

- 1) Otevřu Postmana a zvolím metodu POST
- 2) Přejdu do sekce body a zadám údaje s duplicitním emailem jiného studenta:

```
{  
  "id": 1790,  
  "firstName": "lenka",
```

```
    "lastName": "LENKOVA",  
    "email": "napis@sms.cz",  
    "age": 50  
}
```

3) Stisknu tlačítko send

Očekávaný výsledek: Vypíše se chyba, student s takovým emailem již existuje

Skutečný výsledek: Student se uložil do databáze i s duplicitním emailem

**Abstract:** Vytvoření studenta se záporným věkem

1) Otevřu Postmana a zvolím metodu POST

2) Přejdu do sekce Body a zadám:

```
{  
    "firstName": "Klára",  
    "lastName": "Jelínková",  
    "email": "klara.jelinkova@seznam.com",  
    "age": -100  
}
```

3) Stisknu tlačítko Send

Očekávaný výsledek: Vypíše se chyba, student s takovým věkem nemůže být založen

Skutečný výsledek: Student se uložil i se záporným věkem

**Abstract:** Vytvoření studenta pouze s čísly a speciálními znaky

1) Otevřu Postmana a zvolím metodu POST

2) Přejdu do sekce Body a zadám:

```
{  
    "firstName": 2485,  
    "lastName": 25,  
    "email": "478//!!?",  
    "age": 100  
}
```

3) Stisknu tlačítko Send

Očekávaný výsledek: Vypíše se chyba, studenta nebylo možné vytvořit

Skutečný výsledek: Student se uložil do databáze

**Abstarct:** Vytvoření studenta s chybějícím jménem, příjmením a emailem

1) Otevřu Postmana a zvolím metodu POST

2) Přejdu do sekce Body a zadám:

```
{  
    "firstName": "",  
    "lastName": "",  
    "email": "",  
    "age": 10  
}
```

3) Stisknu tlačítko Send

Očekávaný výsledek: Vypíše se chyba, student nemohl být uložen, chybí povinné údaje

Skutečný výsledek: Student se uložil do databáze

