

FINÁLNÍ PROJEKT

č.1



Autor: Petra Krutílková
Datum: 2. 11. 2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	6
BUG REPORT	8

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: ***** Password: *****
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.

Testovací scénář 1: GET – Získání dat o existujícím studentovi

- **Název testu:** Získání dat o studentovi pomocí metody GET
- **Popis:** Ověří, zda metoda GET správně vrací data o existujícím studentovi podle jeho ID.
- **Požadavek:** GET `http://108.143.193.45:8080/api/v1/students/1011`
- **Vstupní data:**
 - ID studenta, který existuje v databázi (např. `student_id = 1011`)
- **Očekávaný výsledek:**
 - Status kód 200 (OK)
 - JSON objekt obsahující informace o studentovi (`id`, `age`, `first name`, `last name`, `email`)
 - `id: 1011` `age: 30` `email: problEm@sklAvesnici.com`
 `first name: BoŽenKa` `last name: NEUMÍPSÁT`

Testovací scénář 2: GET – Získání dat o neexistujícím studentovi

- **Název testu:** Získání dat o neexistujícím studentovi pomocí metody GET
- **Popis:** Ověří, zda metoda GET správně vrací data o neexistujícím studentovi podle jeho ID.
- **Požadavek:** GET `http://108.143.193.45:8080/api/v1/students/101100`
- **Vstupní data:**
 - ID studenta, který neexistuje v databázi (např. `student_id = 101100`)
- **Očekávaný výsledek:**
 - Status kód 404 (Not found)

Testovací scénář 1: POST – Vložení nového studenta

- **Název testu:** Vložení nového studenta pomocí metody POST
- **Popis:** Ověří, zda metoda POST správně vytváří nový záznam o studentovi.
- **Požadavek:** POST `http://108.143.193.45:8080/api/v1/students/`
- **Vstupní data:**

```
{  
  "firstName": "testPost",  
  "lastName": "posttest",  
  "email": "testposz@yourmail.com",  
  "age": 10  
}
```
- **Očekávaný výsledek:**
 - Status kód 200 (Ok)

Testovací scénář 2: POST – Vytvoření existujícího studenta

- **Název testu:** Vložení existujícího studenta pomocí metody POST
- **Popis:** Ověří, zda metoda POST správně nevytvoří nový záznam o již existujícím studentovi.

- **Požadavek:** POST <http://108.143.193.45:8080/api/v1/students/>

- **Vstupní data:**

```
{  
  "firstName": "@!123",  
  "lastName": "%=",  
  "email": "zavinac@procento.cz",  
  "age": 18  
}
```

- **Očekávaný výsledek:**
 - Status kód 409 (Conflict)/ student již vytvořen

Testovací scénář 1: DELETE – Smazání existujícího studenta

- **Název testu:** Smazání existujícího studenta pomocí metody DELETE
- **Popis:** Ověří, zda metoda DELETE správně smaže záznam o existujícím studentovi.

- **Požadavek:** <http://108.143.193.45:8080/api/v1/students/347>

- **Vstupní data:**

- ID studenta, který má být smazán id = 347

- **Očekávaný výsledek:**
 - Status kód 200 (OK), což indikuje, že student byl úspěšně smazán.

Testovací scénář 2: DELETE – Smazání neexistujícího studenta

- **Název testu:** Smazání neexistujícího studenta pomocí metody DELETE
- **Popis:** Ověří, zda metoda DELETE vrátí správnou chybovou hlášku v případě neexistujícího studenta.

- **Požadavek:** <http://108.143.193.45:8080/api/v1/students/347>

- **Vstupní data:**

- ID studenta, který má být smazán id = 347

- **Očekávaný výsledek:**
 - Status kód 404 (Not Found), což indikuje, že student není v databázi

EXEKUCE TESTŮ

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

Exekuce testu 1: GET – Získání dat o existujícím studentovi

Očekávaný výsledek:

- Status kód 200 (OK)
- JSON objekt obsahující informace o studentovi (id, age first name, last name, email)
- id: 1011 age: 30 email: problEm@sklAvesnici.com
first name: BoŽenKa last name: NEUMÍPSÁT

- **Skutečný výsledek:**

- Status kód 200 (OK)

```
{  
  "id": 1011,  
  "firstName": "BoŽenKa",  
  "lastName": "NEUMÍPSÁT",  
  "email": "problEm@sklAvesnici.com",  
  "age": 30  
}
```

Testovací scénář 2: GET – Získání dat o neexistujícím studentovi

Očekávaný výsledek:

- Status kód 404 (Not found)

- **Skutečný výsledek:**

- Status kód 500 (Internal Server Error)

```
"timestamp": "2024-11-02T14:13:39.288+00:00",  
"status": 500,  
"error": "Internal Server Error",  
"message": "",  
"path": "/api/v1/students/101100"
```

Exekuce testu 1: POST – Vložení nového studenta

- **Očekávaný výsledek:**

- Status kód 200 (Ok)

- **Skutečný výsledek:**

- Status kód 200 (Ok) Student byl přidán do databáze

```
{  
  "id": 2087,  
  "firstName": "testPost",  
  "lastName": "POSTTEST",  
  "email": "testposz@yourmail.com",  
  "age": 10  
}
```

Exekuce testu 2: POST – Vytvoření existujícího studenta

- **Očekávaný výsledek:**
 - Status kód 409 (Conflict)/ student již vytvořen
- **Skutečný výsledek:**
- Status kód 200 (OK) byl vytvořen duplikát pod novým id

```
"id": 2089,  
"firstName": "@!123",  
"lastName": "%=;",  
"email": "zavinac@procento.cz",  
"age": 18
```

Exekuce testu 1: DELETE – Smazání existujícího studenta

Očekávaný výsledek:

- Status kód 200 (OK), což indikuje, že student byl úspěšně smazán.
- **Skutečný výsledek:**
 - Status kód 200 (OK), student byl smazán

Testovací scénář 2: DELETE – Smazání neexistujícího studenta

Očekávaný výsledek:

- Status kód 404 (Not Found), což indikuje, že student není v databázi
- **Skutečný výsledek:**
 - Status kód 500 (Internal Server Error)

BUG REPORT

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

Bug report 2: GET – Získání dat o neexistujícím studentovi

- **Název testu:** Získání dat o neexistujícím studentovi pomocí metody GET
- **Popis:** Ověří, zda metoda GET správně vrací data o neexistujícím studentovi podle jeho ID.

Popis chyby:

Při pokusu o získání dat o neexistujícím studentovi (pomocí metody GET a konkrétního ID) API vrací chybu **500 (Internal Server Error)** namísto očekávané chyby **404 (Not Found)**. Tento problém naznačuje, že aplikace není schopná správně zpracovat dotaz na neexistující záznam a místo toho vrací interní chybu serveru.

Kroky k reprodukci chyby:

1. Otevřete Postman a vyberte metodu GET.
2. Zadejte URL: `http://108.143.193.45:8080/api/v1/students/101100`
3. Ujistěte se, že ID studenta (101100) neexistuje v databázi.
4. Odešlete požadavek.

Očekávaný výsledek:

- Status kód **404 (Not Found)**.
- Zpráva označující, že záznam nebyl nalezen, případně žádná data v těle odpovědi.

Skutečný výsledek:

- Status kód **500 (Internal Server Error)**.

Bug report 2: POST – Vytvoření existujícího studenta

- **Název testu:** Vložení existujícího studenta pomocí metody POST
- **Popis:** Ověří, zda metoda POST správně nevytvoří nový záznam o již existujícím studentovi.

Popis chyby:

Při pokusu o vytvoření již existujícího studenta pomocí metody POST API vrací status kód **200 (OK)** a vytváří duplicitní záznam s novým ID namísto očekávaného status kódu **409 (Conflict)**. Tento výsledek naznačuje, že server nesprávně umožňuje vytváření identických záznamů bez kontroly duplicit.

Kroky k reprodukci chyby:

1. Otevřete Postman a vyberte metodu POST.
2. Zadejte URL: `http://108.143.193.45:8080/api/v1/students/`.
3. Do sekce **Body** vložte následující JSON data:

```
{
  "firstName": "@!123",
  "lastName": "%=",
  "email": "zavinac@procento.cz",
  "age": 18
}
```


4. Odešlete požadavek jednou, abyste vytvořili záznam o studentovi, a následně jej odešlete znovu se stejnými údaji.

Očekávaný výsledek:

- Status kód **409 (Conflict)** s chybovou zprávou indikující, že student s těmito údaji již existuje v databázi a není možné vytvořit duplicitní záznam.

Skutečný výsledek:

- Status kód **200 (OK)**.
- Nový záznam o studentovi je vytvořen se stejnými údaji, ale s novým ID:
- Status kód 200 (OK) byl vytvořen duplikát pod novým id

```
"id": 2089,  
"firstName": "@!123",  
"lastName": "%=",  
"email": "zavinac@procento.cz",  
"age": 18
```

Bug report 2: DELETE – Smazání neexistujícího studenta

- **Název testu:** Smazání neexistujícího studenta pomocí metody DELETE
- **Popis:** Ověří, zda metoda DELETE vrátí správnou chybovou hlášku v případě neexistujícího studenta.

Popis chyby:

Při pokusu o smazání neexistujícího studenta (pomocí metody DELETE a konkrétního ID) API vrací chybu **500 (Internal Server Error)** namísto očekávané chyby **404 (Not Found)**. Tato chyba naznačuje, že server nemá správnou logiku pro zachycení pokusu o smazání neexistujícího záznamu, což vede k nesprávné odpovědi.

Kroky k reprodukcí chyby:

1. Otevřete Postman a vyberte metodu DELETE.
2. Zadejte URL: `http://108.143.193.45:8080/api/v1/students/347`.
3. Ujistěte se, že student s ID 347 neexistuje v databázi.
4. Odešlete požadavek.

Očekávaný výsledek:

- Status kód **404 (Not Found)** s chybovou zprávou indikující, že student s daným ID nebyl nalezen.

Skutečný výsledek:

- Status kód **500 (Internal Server Error)**, což naznačuje interní chybu serveru namísto správného zachycení neexistujícího záznamu.