

# FINÁLNÍ PROJEKT

## č.1



Autor: Anna Šupová  
Datum: 15.5.2025

## OBSAH

<b>ZADÁNÍ</b>	<b>3</b>
<b>TESTOVACÍ SCÉNÁŘE</b>	<b>4-6</b>
<b>BUG REPORT</b>	<b>7</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	<b>Default scheme:</b> qa_demo <b>Host:</b> aws.connect.psdb.cloud <b>Port:</b> 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 A EXEKUCE TESTŮ

Na základě uvedených testovacích scénářů jsem ověřila funkčnost aplikace.

ID	Název	Kroky	Očekávaný výsledek	Skutečný výsledek	Výsledek testu
1	Metoda GET s validními údaji	<p>V Postmanu jsem nastavila GET metodu. Do URL zadala <code>https://students-api.engeto.com/api/v1/students/1</code>. Poslala jsem požadavek. Zkontrolovala jsem, že status kód je 200 OK. Porovnala jsem data v odpovědi s databází (MySQL) – ID 1 existuje, jméno, příjmení, email a věk odpovídají.</p>	Status kód: 200 OK	<pre>status kód 200 - {   "id": 1,   "firstName":     "Peter",   "lastName":     "VIANOCNY",   "email":     "peter30@yourmail.com",   "age": 19 }</pre>	✓
2	Metoda GET s nevalidními údaji	<p>V Postmanu jsem nastavila GET metodu. Do URL zadala <code>https://students-api.engeto.com/api/v1/students/420</code>. Poslala požadavek. Očekávala jsem status 404 Not Found, protože ID 420 v databázi není. Místo toho přišel status 500 Internal Server Error. Prohlédla jsem si chybu v odpovědi (chybí konkrétní zpráva).</p>	Status kód: 404 Not Found	<pre>status kód 500 internal server error {   "timestamp":     "2025-05-17T10:3 4:52.229+00:00",   "status": 500,   "error": "Internal Server Error",   "message": "",   "path":     "/api/v1/students /420%0A" }</pre>	✗
3	Metoda POST s validními daty	<p>V Postmanu jsem zvolila POST metodu. Nastavila URL <code>https://students-api.engeto.com/api/v1/students/</code>. Do těla requestu vložila <pre>{   "firstName": "anicka",   "lastName": "supova",   "email": "sigrspyco@gmail.com",   "age": 25 }</pre> Poslala požadavek. Očekávala status 201 Created (nový záznam vytvořen).</p>	Status kód: 201 Created	<pre>status kód 200 OK {   "id": 142,   "firstName":     "anicka",   "lastName":     "SUPOVA",   "email":     "sigrspyco@gmail .com",   "age": 25 }</pre>	✗

		Skutečně přišel status 200 OK s daty nového studenta v odpovědi.			
4	Metoda POST s chybějícím parametrem	<p>V Postmanu jsem zvolila POST metodu. Nastavila URL, hlavičky i JSON tělo bez firstName.</p> <pre>{   "lastName": "supova1",   "email": "sigrspyco1@gmail.com",   "age": 25 }</pre> <p>Poslala požadavek. Očekávala jsem chybu 400 Bad Request, protože firstName je povinný. Místo toho přišel status 500 Internal Server Error.</p>	Status kód: 400 Bad Request	<pre>status kód 500 internal server error {   "timestamp": "2025-05-17T10:5 6:53.599+00:00",   "status": 500,   "error": "Internal Server Error",   "message": "",   "path": "/api/v1/students /" }</pre>	✗ failed
5	Metoda POST s parametrem navíc	<p>V Postmanu jsem zvolila POST metodu. Přidala jsem do JSON těla POST requestu nepovolený parametr identifier.</p> <pre>{   "identifier": 5   "firstName": "anicka1",   "lastName": "supova1",   "email": "sigrspyco1@gmail.com",   "age": 25 }</pre> <p>Nastavila URL a hlavičky správně. Poslala požadavek. Očekávala jsem status 400 Bad Request (protože parametr není validní). Skutečný status byl také 400.</p>	Status kód: 400 Bad Request	<pre>status kód - 400 bad request {   "timestamp": "2025-05-17T10:5 9:30.529+00:00",   "status": 400,   "error": "Bad Request",   "message": "",   "path": "/api/v1/students /" }</pre>	✓
6	Metoda POST se speciálními znaky	<p>V Postmanu jsem zvolila POST metodu. Do firstName jsem vložila speciální znaky</p> <pre>{   "firstName": "anicka1@{Đ",   "lastName": "supova1",   "email": "sigrspyco1@gmail.com",   "age": 25 }</pre> <p>Vytvořila POST request s tímto tělem. Poslala požadavek na správnou URL. Očekávala jsem chybu 400 Bad Request (neplatné znaky). API vrátilo status 200 OK a záznam vytvořilo.</p>	Status kód: 400 Bad Request	<pre>status kód 200 OK {   "id": 144,   "firstName": "anicka1@{Đ",   "lastName": "SUPOVA1",   "email": "sigrspyco1@gm ail.com",   "age": 25 }</pre>	✗

7	Metoda POST se špatným formátem emailu	<p>V Postmanu jsem zvolila POST metodu. Do emailu jsem zadala nevalidní formát.</p> <pre>{   "firstName": "anicka1@{Đ",   "lastName": "supova1",   "email": "anicka",   "age": 25 }</pre> <p>Poslala POST request se špatným emaillem.</p> <p>Očekávala jsem status 400 Bad Request. Přišel status 200 OK a záznam byl vytvořen.</p>	Status kód: 400 Bad Request	<pre>-status kód 200 OK {   "id": 145,   "firstName": "anicka1@{Đ",   "lastName": "SUPOVA1",   "email": "anicka",   "age": 25 }</pre>	✗
8	Metoda DELETE existující záznam	<p>V Postmanu jsem zvolila DELETE metodu. Nastavila DELETE request na URL s existujícím ID</p> <p><a href="https://students-api.engeto.com/api/v1/students/145">https://students-api.engeto.com/api/v1/students/145</a></p> <p>Poslala požadavek.</p> <p>Očekávala jsem status 200 OK (úspěšné smazání).</p> <p>Status 200 také přišel.</p>	Status kód: 200 OK	status kód 200 OK	✓
9	Metoda DELETE neexistující záznam	<p>V Postmanu jsem zvolila DELETE metodu. Nastavila DELETE request na neexistující ID</p> <p><a href="https://students-api.engeto.com/api/v1/students/420">https://students-api.engeto.com/api/v1/students/420</a></p> <p>Poslala požadavek.</p> <p>Očekávala status 404 Not Found. Místo toho přišel status 500 Internal Server Error.</p>	Status kód: 404 Not Found	<pre>status kód 500 internal server error {   "timestamp": "2025-05-17T11:11:30.910+00:00",   "status": 500,   "error": "Internal Server Error",   "message": "",   "path": "/api/v1/students/420" }</pre>	✗

# BUG REPORT

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

bug	probability	severity	test case number	mitigace
GET - neexistující student	vždy	vysoká	2	Odpověď serveru by měla být 404 not found
POST vytvoření nového studenta s validními údaji	vždy	normal	3	Odpověď serveru by měla být 201 created
POST - vytvoření studenta s chybějícím parametrem v body requestu	vždy	vysoká	4	Odpověď serveru by měla být 400 bad request
POST - vytvoření studenta se speciálním znakem ve jméně	vždy	vysoká	6	Odpověď serveru by měla být 400 bad request
POST - vytvoření studenta bez znaku @ v emailové adrese	vždy	vysoká	7	Odpověď serveru by měla být 400 bad request
DELETE - smazání neexistujícího studenta	vždy	normal	9	Odpověď by měla být 404 not found