

FINÁLNÍ PROJEKT

č. 1



Autor: Bc. Štěpánka Klofáčová
Datum: 17. 2. 2025

OBSAH

ZADÁNÍ	3
EXEKUCE TESTŮ	4
TESTOVACÍ SCÉNÁŘE	5
BUG REPORT	7

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.

EXEKUCE TESTŮ

Testovací scénáře jsem provedla, přikládám výsledky testů.

TESTOVACÍ SCÉNÁŘE

Na základě uvedených testovacích scénářů jsem ověřila funkčnost aplikace.

Test č.	Popis testu	Reprodukce testu	Očekávaný výsledek	Reálný výsledek	Poznámky
1	hledání existujících studentů dle jejich ID	ID: 437, 1200, 2696	studenti nalezeni + status kód 200	studenti nalezeni + status kód 200	ověřeno v databázi přes MySQL
3	hledání studentů dle ID v neplatném formátu	ID: osm, 0sm, 26x, 26*, -1	studenti nenalezeni + status kód 400	studenti nenalezeni + status kód 400	
6	vytvoření nového studenta s údaji v platném formátu	{ "firstName": "Tris", "lastName": "Prior", "email": "tris@gmail.com", "age": 16 }	student vytvořen, přiděleno ID + status kód 200	student vytvořen, přiděleno ID + status kód 200	ověřeno v databázi přes MySQL
8	vymazání existujících studentů dle jejich existujících ID	ID: 2762, 2792, 2793	student smazán + status kód 200	student smazán + status kód 200	byl očekáván i kód 204, protože je pro metodu Delete typičtější (když je požadavek úspěšný způsobem, že odpověď neobsahuje žádná data)
10	vymazání neexistujících studentů dle ID v neplatném formátu	ID: šest, š3st, 25x, 25*, -6, -3000	studenti nenalezeni + status kód 400	studenti nenalezeni + status kód 400	u minusových ID se vrací status kód 500
11	vytvoření nového studenta bez vyplnění osobních údajů	{ "firstName": " ", "lastName": " ", "email": " ", "age": }	student nevytvořen + status kód 400	student nevytvořen + status kód 400	

12	vytvoření nového studenta s vyplněným pouze vždy jedním řádkem („firstName“, „lastName“ nebo „email“)	vyplněný pouze řádek „firstName“: { "firstName": "Tris", "lastName": " ", "email": " ", "age": }	student nevytvořen + status kód 400	student nevytvořen + status kód 400	(řádek „age“ úmyslně vynechán pro pozdější testy)
15	vytvoření nového studenta s vyplněnými vždy dvěma řádky (kombinace řádků „firstName“, „lastName“, „email“)	kombinace řádků „firstName“ a „email“: { "firstName": "Tris", "lastName": " ", "email": "tris@gmail.com", "age": }	student nevytvořen + status kód 400	student nevytvořen + status kód 400	tento výsledek je mírně překvapivý (viz test č. 14), kombinace těchto údajů je obecně relevantnější než věk
19	vytvoření nového studenta s platnými údaji - kromě řádku „age“, kde jsou použity různé znaky kromě samostatných celých čísel	{ "firstName": "Zeke", "lastName": "PODRAG", "email": "zeke@p.cz", "age": "7teen" }	student nevytvořen + status kód 400	student nevytvořen + status kód 400	vyzkoušeny i tyto varianty: 7 7, 7 teen, s7v7n, \$ß, ščř, @, {6000}, <6000>, [60]
22	změna všech údajů existujícího studenta	změna e-mailu: { "id": 2795, "firstName": "Tobias", "lastName": "EATON", "email": "for4@yahoo.com", "age": 18 }	údaje změněny + status kód 200	údaje změněny + status kód 200	ověřeno v databázi přes MySQL; tento výsledek je mírně překvapivý, protože změnu existujícího záznamu by měla primárně zastávat spíše metoda Put než Post
25	rychlost aplikace	do řádku „firstName“ a „lastName“ bylo zapsáno celkem 4362 znaků	student vytvořen za méně než 1s, přiděleno ID + status kód 200	student vytvořen za 0,396s, přiděleno ID + status kód 200	

BUG REPORT

Na základě provedených scénářů jsem objevila uvedené chyby aplikace.

Test č.	Popis testu	Reprodukce chyby	Očekávaný výsledek	Reálný výsledek	Důležitost	Poznámky
2	hledání neexistujících studentů dle neexistujících ID v platném formátu	ID: 0, 1, 344	studenti nenalezeni + status kód 404	studenti nenalezeni + status kód 500	střední	chybný požadavek by měl ukazovat spíše na chybu klienta (404) než na chybu serveru (500)
4	hledání existujících studentů dle jejich existujících, avšak upravených ID	ID: 000437, 0001200, 0002696	studenti nenalezeni + status kód 404	studenti nalezeni + status kód 200	nízká	nuly na začátku ID by sice neměly napáchat žádné škody, nicméně pro nezkušené uživatele by mohly být matoucí
5	hledání existujících studentů dle jejich existujících ID, avšak v neplatném formátu	ID: 4 3 7, 12 00, 2 69 6	studenti nenalezeni + status kód 400	studenti nalezeni + status kód 200	střední	hledání v databázi přes MySQL vrátilo error
7	vymazání neexistujících studentů dle neexistujících ID v platném formátu	ID: 344, 444, 599	studenti nenalezeni + status kód 400	studenti nenalezeni + status kód 500	střední	chybný požadavek by měl ukazovat spíše na chybu klienta (400) než na chybu serveru (500)
9	vytvoření nového studenta s údaji v platném formátu s vynecháním vždy jednoho či více řádků	vynechání řádku „email“: { "firstName": "Tris", "lastName": "Prior", "age": 16 }	student nevytvořen + status kód 400	student nevytvořen + status kód 500	střední	chybný formát požadavku by měl ukazovat spíše na chybu klienta (400) než na chybu serveru (500)
13	vytvoření nového studenta pouze s vyplněným řádkem „age“	{ "firstName": " ", "lastName": " ", "email": " ", "age": 16 }	student nevytvořen + status kód 400	student vytvořen, přiděleno ID + status kód 200	vysoká	ověřeno v databázi přes MySQL

		"age": 16 }				
14	vytvoření nového studenta pouze s vyplněnými řádky „firstName“ a „lastName“	{ "firstName": "Tris", "lastName": "Prior", "email": " ", "age": }	student vytvořen, přiděleno ID + status kód 200	student nevytvořen + status kód 400	nízká	tento výsledek je překvapivý, protože z osobní zkušenosti bych předpokládala, že je důležitější studenta alespoň založit (např. pro info, kolik studentů je reálně v daném kurzu) a jeho ostatní údaje doplnit později, než ho nezapsat vůbec
16	vytvoření nového studenta s vyplněnými vždy dvěma řádky, a to v kombinaci s řádkem „age“	kombinace řádků „email“ a „age“: { "firstName": " ", "lastName": " ", "email": "tris@gmail.com", "age": 16 }	student nevytvořen + status kód 400	student vytvořen, přiděleno ID + status kód 200	vysoká	zdá se, že jakmile údaje o novém studentovi obsahují údaj o věku, je student vždy vytvořen, přitom je to ten nejméně cenný údaj (každý rok se stejně mění)
17	vytvoření nového studenta s platnými údaji - kromě řádku „age“, který zůstane nevyplněn	{ "firstName": "Tobias", "lastName": "Eaton", "email": "44@yahoo.com", "age": }	student vytvořen, přiděleno ID + status kód 200	student nevytvořen + status kód 400	střední	překvapivý výsledek, viz test č. 14 a 16
18	vytvoření nového studenta s platnými údaji - kromě řádku „age“, kde jsou použita desetinná čísla	řádek „age“ obsahující desetinné číslo místo celého: { "firstName": "Tobias", "lastName": "Eaton", "email": "four@yahoo.com", "age": 18.555 }	student nevytvořen + status kód 400	student vytvořen s opraveným údajem o věku (jen „18“), přiděleno ID + status kód 200	nízká	nejde vlastně ani tak o bug jako o správné chování systému, které v kolonce „age“ zachovává pouze tu část čísla, která je před desetinnou čárkou

		}				
20	vytvoření nového studenta s platnými údaji - kromě řádku „age“, kde jsou použita záporná čísla	{ "firstName": "Zeke", "lastName": "PODRAG", "email": "zeke@p.cz", "age": -12 }	studenti nevytvoření + status kód 400	studenti vytvoření, přiděleno ID + status kód 200	vysoká	ověřeno v databázi přes MySQL; u věku je z podstaty věci očekáváno vždy kladné číslo
21	vytvoření nového studenta s platnými údaji - kromě řádku „age“, kde jsou použita příliš vysoká čísla	{ "firstName": "Ezekiel", "lastName": "PODRAG", "email": "zeke@p.cz", "age": 6000 }	studenti nevytvoření + status kód 400	studenti vytvoření, přiděleno ID + status kód 200	vysoká	ověřeno v databázi přes MySQL; v kolonce věku se z podstaty věci očekává omezení maximální délkou života člověka
23	vytvoření nového studenta s platnými údaji - kromě řádku „email“, kde jsou použity neplatné tvary e-mailové adresy	{ "firstName": "Tori", "lastName": "WU", "email": "ToriWu.wz.cz", "age": 45 }	studenti nevytvoření + status kód 400	studenti vytvoření, přiděleno ID + status kód 200	střední	ověřeno v databázi přes MySQL; vyzkoušeny i tyto varianty: @tori.cz, @@@, >>šakal@, "mejl vůbec nemá", \$\$ßß××÷÷<<>>
24	vytvoření nového studenta s platnými údaji - kromě řádků „firstName“ a „lastName“	{ "firstName": "3v3lyn", "lastName": "JOHN5ON", "email": "ev@johns.on", "age": 48 }	studenti nevytvoření + status kód 400	studenti vytvoření, přiděleno ID + status kód 200	vysoká	ověřeno v databázi přes MySQL; matriční zákony nepovolují ve jméně ani příjmení čísla ani jiné speciální znaky (případně je nutná lokalizace); vyzkoušeny kombinace, které obsahovaly pouze čísla či speciální znaky
26	omezení maximálního počtu znaků v jednotlivých řádcích	do řádku „firstName“ a „lastName“ bylo zapsáno celkem 7477 znaků	student nevytvořen + status kód 400	student vytvořen, přiděleno ID + status kód 200	střední	ověřeno v databázi přes MySQL