

FINÁLNÍ PROJEKT

č.1



ENGETO

Autor: Jitka Šafářová
Datum: 10.6.2024

OBSAH

ZADÁNÍ	3
Přístupové údaje:	3
Poznámky:	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	9
BUG REPORT	15

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.

Doplnění zadání:

1. všechny atributy/údaje jsou povinné
2. všechny atributy se mají uložit do databáze pomocí lowerCase() pravidla (všechna velká písmena se změní na malá)

TESTOVACÍ SCÉNÁŘE

Na základě uvedených testovacích scénářů jsem ověřila funkčnost aplikace:

Metoda GET:

TC ID	Abstrakt	Kroky	Očekávaný výsledek
1.	Zjištění informací o existujícím studentovi (validní ID)	1. V nástroji Postman zadat GET požadavek na endpoint: <code>http://108.143.193.45:8080/api/v1/students/490</code> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 200 OK 2. Tělo odpovědi zobrazí informace o studentovi ve správném formátu: { " id": 490, " firstName": "Bilbo", " lastName": "PYTLÍK", " email": "nemam", " age": 100 } 3. Tyto informace odpovídají informacím uloženým v databázi

TC ID	Abstrakt	Kroky	Očekávaný výsledek
2.	Zjištění informací o neexistujícím studentovi (nevalidní ID)	1. V nástroji Postman zadat GET požadavek na endpoint: <code>http://108.143.193.45:8080/api/v1/students/197</code> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 404 Not Found 2. Zobrazí se hláška, která upozorní, že student se zadaným ID neexistuje.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
3.	Zjištění informací o studentovi s neplatným formátem ID – zadání jména místo ID	1. V nástroji Postman zadat GET požadavek na endpoint: <code>http://108.143.193.45:8080/api/v1/students/Bilbo</code> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 400 Bad Request 2. Zobrazí se upozornění na chybně zadaný požadavek.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
4.	Zjištění informací o studentovi – nezadání hodnoty (ID)	1. V nástroji Postman zadat GET požadavek na endpoint: <code>http://108.143.193.45:8080/api/v1/students/</code> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 4** 2. Zobrazí se upozornění na chybně zadaný požadavek. Aplikace neumožní zobrazení jakýchkoli informací o studentovi/studentech bez zadání validního ID.

Metoda POST:

TC ID	Abstrakt	Kroky	Očekávaný výsledek
5.	Ověření, zda se při založení nového studenta údaje o studentovi ukládají do databáze pomocí lower case pravidla	1. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "firstName": "Harry", "lastName": "Potter", "age": 11, "email": "harrypotter@hogwarts.uk" }</pre> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 201 Created 2. Nový student se správně založil a všechny údaje jsou malými písmeny. <pre>{ "id": 1140, "firstName": "harry", "lastName": "potter", "email": "harrypotter@hogwarts.uk", "age": 11 }</pre> 3. Uvedené údaje se totožně zobrazují i v databázi.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
6.	Založení studenta s vynecháním parametru „firstName“ (křestního jména)	1. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "lastName": "Weasley", "age": 12, "email": "fredweasley@hogwarts.uk" }</pre> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 400 Bad Request 2. Odpověď s chybovou zprávou o chybějícím parametru.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
7.	Založení studenta s prázdným polem „firstName“ (křestní jméno)	1. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "firstName": "", "lastName": "Potter", "age": 11, "email": "harrypotter@hogwarts.uk" }</pre> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 400 Bad Request 2. Odpověď obsahuje chybovou zprávu o nevyplněném poli.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
8.	Založení studenta s nevalidním e-mailem	1. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "firstName": "Hermiona", "lastName": "Granger", "age": 11, "email": "alohomora" }</pre> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 400 Bad Request 2. Odpověď obsahuje chybovou zprávu o neplatném formátu e-mailu.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
9.	Založení studenta s nečíselnou hodnotou pro věk	1. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "firstName": "Ron", "lastName": "Weasley", "age": „jedenáct“, "email": "ronweasley@hogwarts.uk" }</pre> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
10.	Založení studenta s duplicitním e-mailem	1. Ověřit, že v databázi je již založen student s e-mailem "harrypotter@hogwarts.uk" 2. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "firstName": "Draco", "lastName": "Malfoy", "age": 11, "email": "harrypotter@hogwarts.uk" }</pre> 3. Kliknout na tlačítko „Send“.	1. Stavový kód: 4** 2. Odpověď zobrazí chybovou zprávu – student s touto e-mailovou adresou již existuje.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
11.	Založení studenta se speciálními znaky v poli "firstName" a "lastName"	1. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "firstName": "Neville\$", "lastName": "Longbottom\$", "age": 11, "email": "neville@hogwarts.uk" }</pre> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
12.	Založení studenta s parametrem navíc	1. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "firstName": "Neville", "lastName": "Longbottom", "age": 11, "email": "neville@hogwarts.uk" "parametrNavic": "Neville je smolař" }</pre> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
13.	Založení studenta s věkem vyjádřeným desetinným číslem	1. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "firstName": "Neville", "lastName": "Longbottom", "age": 11.5 "email": "neville@hogwarts.uk" }</pre> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
14.	Založení studenta s věkem vyjádřeným záporným číslem	1. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "firstName": "Neville", "lastName": "Longbottom", "age": -11, "email": "neville@hogwarts.uk" }</pre> 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.

Metoda DELETE:

TC ID	Abstrakt	Kroky	Očekávaný výsledek
15.	Smazání existujícího studenta	1. V nástroji Postman zadat POST požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ za použití následujících údajů: <pre>{ "firstName": "Severus", "lastName": "Snape", "age": 31, "email": "severus.snape@hogwarts.uk" }</pre> 2. Kliknout na tlačítko „Send“. 3. Zaznamenat si ID vytvořeného studenta. 4. Ověřit v databázi, že se student správně založil. 5. V nástroji Postman zadat DELETE požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/{id}, kde {id} je ID získané v kroku č.3. 6. Kliknout na tlačítko „Send“.	Po dokončení kroku č.6: 1. Stavový kód: 200 OK 2. Záznam týkající se studenta s dotčeným ID je z databáze smazán.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
16.	Smazání neexistujícího studenta	1. Ověřit v databázi, že student s ID 1184 neexistuje. 2. V nástroji Postman zadat DELETE požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/1184 3. Kliknout na tlačítko „Send“.	1. Stavový kód: 404 Not Found 2. Tělo odpovědi zobrazí odpovídající zprávu.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
17.	Smazání záznamu s neplatným formátem ID studenta	1. V nástroji Postman zadat DELETE požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/abc 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 400 Bad Request 2. Tělo odpovědi zobrazí odpovídající zprávu.

TC ID	Abstrakt	Kroky	Očekávaný výsledek
18.	Smazání záznamu bez zadání ID studenta	1. V nástroji Postman zadat DELETE požadavek na endpoint: http://108.143.193.45:8080/api/v1/students/ 2. Kliknout na tlačítko „Send“.	1. Stavový kód: 405 Method Not Allowed 2. Tělo odpovědi zobrazí odpovídající zprávu.

EXEKUCE TESTŮ

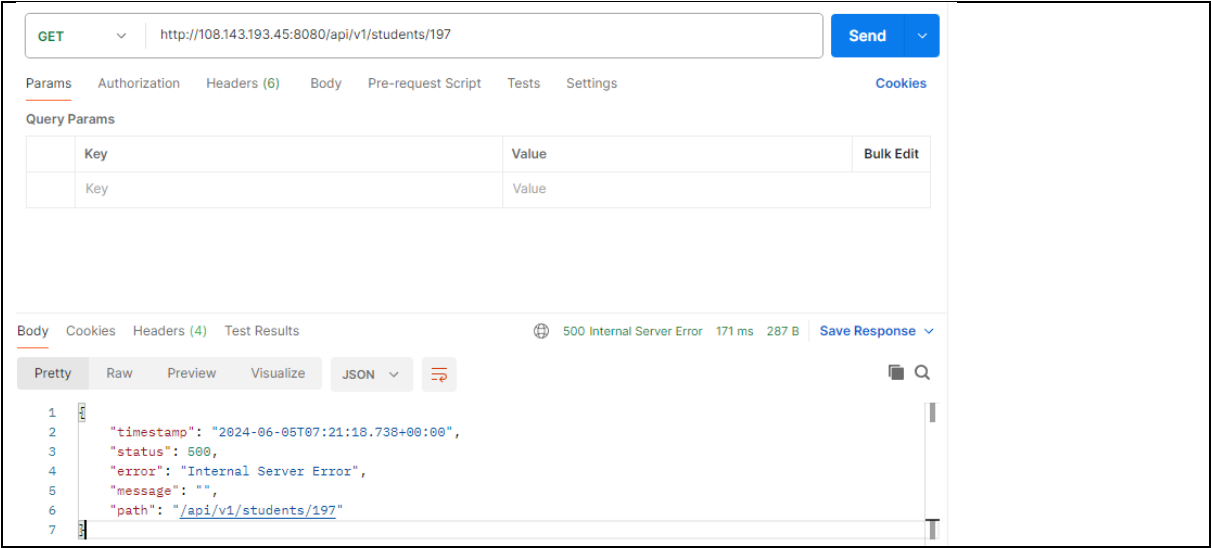
Metoda GET:

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
1.	ANO	<div>1. Stavový kód: 200 OK</div> <div>2. Tělo odpovědi zobrazí informace o studentovi ve správném formátu:</div> <div><pre>{ "id": 490, "firstName": "Bilbo", "lastName": "PYTLÍK", "email": "nemam", "age": 100 }</pre></div> <div>3. Tyto informace odpovídají informacím uloženým v databázi</div>	<div>1. Stavový kód: 200 OK</div> <div>2. Tělo odpovědi zobrazí informace o studentovi ve správném formátu:</div> <div><pre>{ "id": 490, "firstName": "Bilbo", "lastName": "PYTLÍK", "email": "nemam", "age": 100 }</pre></div> <div>3. Tyto informace odpovídají informacím uloženým v databázi</div>

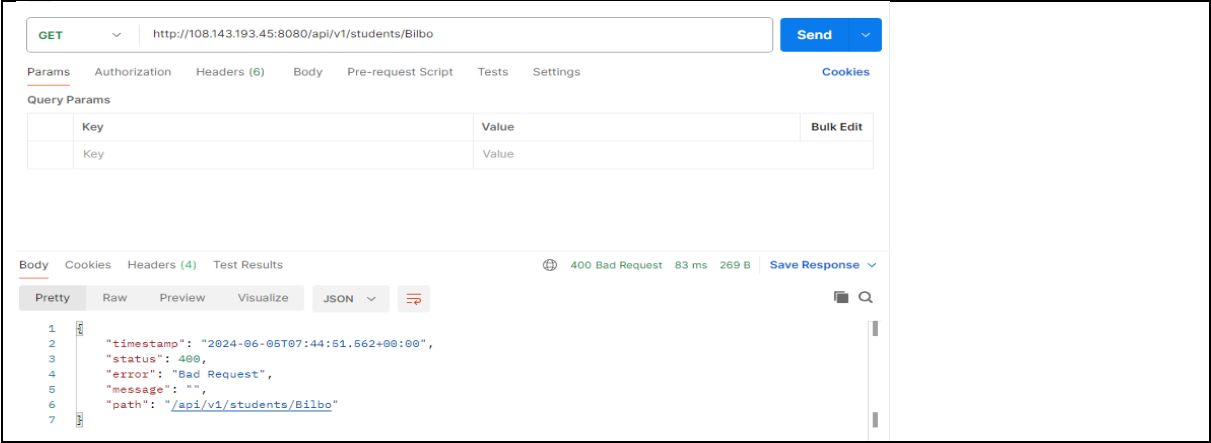
	id	age	email	first_name	last_name
▶	490	100	nemam	Bilbo	PYTLÍK
•	NULL	NULL	NULL	NULL	NULL

(stav v databázi)

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
2.	NE	1. Stavový kód: 404 Not Found 2. Zobrazí se hláška, která upozorní, že student se zadaným id neexistuje.	1. Stavový kód: 500 Internal Server Error 2. { "timestamp": "2024-06-05T07:21:18.738+00:00", "status": 500, "error": "Internal Server Error", "message": "", "path": "/api/v1/students/197" }



TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
3.	ANO	1. Stavový kód: 400 Bad Request 2. Zobrazí se upozornění na chybně zadaný požadavek.	1. Stavový kód: 400 Bad Request 2. { "timestamp": "2024-06-05T07:44:51.562+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/Bilbo" }



TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
4.	NE	1. Stavový kód: 4** 2. Zobrazí se upozornění na chybně zadaný požadavek. Aplikace neumožní zobrazení jakýchkoli informací o studentovi/studentech bez zadání validního ID.	1. Stavový kód: 200 OK 2. Zobrazí se kompletní záznamy o všech studentech.

Metoda POST:

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
5.	NE	1. Stavový kód: 201 Created 2. Nový student se správně založil a všechny údaje jsou malými písmeny. <pre>{ "id": 1140, "firstName": "harry", "lastName": "potter", "email": "harrypotter@hogwarts.uk", "age": 11 }</pre> 3. Uvedené údaje se totožně zobrazují i v databázi.	1. Stavový kód: 200 OK 2. Nový student se založil, ale velká písmena se nepřevédla na malá. U parametru „lastName“ se dokonce všechna písmena chybně převédla na velká. <pre>{ "id": 1140, "firstName": "Harry", "lastName": "POTTER", "email": "harrypotter@hogwarts.uk", "age": 11 }</pre> 3. Uvedené údaje se totožně zobrazují i v databázi.

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
6.	NE	1. Stavový kód: 400 Bad Request 2. Odpověď s chybovou zprávou o chybějícím poli.	1. Stavový kód: 500 Internal Server Error 2. Odpověď: <pre>{ "timestamp": "2024-06-05T12:28:53.628+00:00", "status": 500, "error": "Internal Server Error", "message": "", "path": "/api/v1/students/" }</pre>

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
7.	NE	1. Stavový kód: 400 Bad Request 2. Odpověď s chybovou zprávou o nevyplněném poli.	1. Stavový kód: 200 OK 2. Založí se student s prázdným křestním jménem: <pre>{ "id": 1144, "firstName": "", "lastName": "POTTER", "email": "harrypotter@hogwarts.uk", "age": 11 }</pre>

	id	age	email	first_name	last_name
▶	1144	11	harrypotter@hogwarts.uk		POTTER
*	NULL	NULL	NULL	NULL	NULL

(kontrola v databázi)

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
8.	NE	1. Stavový kód: 400 Bad Request 2. Odpověď obsahuje chybovou zprávu o neplatném formátu e-mailu.	1. Stavový kód: 200 OK 2. Založí se student s neplatným e-mailem: { "id": 1146, "firstName": "Hermiona", "lastName": "GRANGER", "email": "alohomora", "age": 11 }

	id	age	email	first_name	last_name
▶	1146	11	alohomora	Hermiona	GRANGER
*	NULL	NULL	NULL	NULL	NULL

(kontrola v databázi)

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
9.	ANO	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.	1. Stavový kód: 400 Bad Request 2. Odpověď: <pre>{ "timestamp": "2024-06-05T13:09:06.327+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/" }</pre>

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
10.	NE	1. Stavový kód: 4** 2. Odpověď zobrazí chybovou zprávu – student s touto e-mailovou adresou již existuje.	1. Stavový kód: 200 OK 2. Student se založí, viz screen

	id	age	email	first_name	last_name
▶	1140	11	harrypotter@hogwarts.uk	Harry	POTTER
	1142	11	harrypotter@hogwarts.uk		POTTER
	1144	11	harrypotter@hogwarts.uk		POTTER
	1152	11	harrypotter@hogwarts.uk	Draco	MALFOY
⬇	NULL	NULL	NULL	NULL	NULL

(kontrola v databázi)

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
11.	NE	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.	1. Stavový kód: 200 OK 2. Odpověď: <pre>{ "id": 1153, "firstName": "Neville\$", "lastName": "LONGBOTTOM\$", "email": "neville@hogwarts.uk", "age": 11 }</pre>

	id	age	email	first_name	last_name
▶	1153	11	neville@hogwarts.uk	Neville\$%	Longbottom\$%
*	NULL	NULL	NULL	NULL	NULL

(kontrola v databázi)

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
12.	ANO	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.
<div><div>POST</div><div>http://108.143.193.45:8080/api/v1/students/</div><div>Send</div></div> <div>ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings</div> <div>noneform-datax-www-form-urlencodedrawbinaryJSON</div> <div><pre>1 { 2 "firstName": "Neville", 3 "lastName": "Longbottom", 4 "age": 11, 5 "email": "neville@hogwarts.uk" 6 "parametrNavic": "Neville je smolař" 7 } 8</pre></div> <div>bodyCookiesHeaders (4)Test Results</div> <div>400 Bad Request80 ms264 BSave Response</div> <div>PrettyRawPreviewVisualizeJSON</div> <div><pre>1 { 2 "timestamp": "2024-06-06T16:58:46.752+00:00", 3 "status": 400, 4 "error": "Bad Request", 5 "message": "", 6 "path": "/api/v1/students/" 7 }</pre></div>			

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
13.	NE	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.	1. Stavový kód: 200 OK 2. Student se založí, datový typ FLOAT se automaticky převede na INTEGER. Odpověď: { " id ": 1154, " firstName ": "Neville", " lastName ": "LONGBOTTOM", " email ": "neville@hogwarts.uk", " age ": 11 }

	id	age	email	first_name	last_name
▶	1154	11	neville@hogwarts.uk	Neville	LONGBOTTOM
*	NULL	NULL	NULL	NULL	NULL

(kontrola v databázi)

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
14.	NE	1. Stavový kód: 400 Bad Request 2. Odpověď zobrazí chybovou zprávu.	1. Stavový kód: 200 OK 2. Student se založí se záporným věkem. Odpověď: <pre>{ "id": 1155, "firstName": "Neville", "lastName": "LONGBOTTOM", "email": "neville@hogwarts.uk", "age": -11 }</pre>

	id	age	email	first_name	last_name
▶	1155	-11	neville@hogwarts.uk	Neville	LONGBOTTOM
*	NULL	NULL	NULL	NULL	NULL

(kontrola v databázi)

Metoda DELETE:

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
15.	ANO	Po dokončení kroku č.6: 1. Stavový kód: 200 OK 2. Záznam týkající se studenta s dotčeným ID je z databáze smazán.	Po dokončení kroku č.6: 1. Stavový kód: 200 OK 2. Záznam týkající se studenta s dotčeným ID je z databáze smazán.

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
16.	NE	1. Stavový kód: 404 Not Found 2. Tělo odpovědi zobrazí odpovídající zprávu.	1. Stavový kód: 500 Internal Server Error 2. Odpověď: <pre>{ "timestamp": "2024-06-10T15:38:55.880+00:00", "status": 500, "error": "Internal Server Error", "message": "", "path": "/api/v1/students/1184" }</pre>

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
17.	ANO	1. Stavový kód: 400 Bad Request 2. Tělo odpovědi zobrazí odpovídající zprávu.	1. Stavový kód: 400 Bad Request 2. Odpověď: <pre>{ "timestamp": "2024-06-10T15:47:19.763+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/abc" }</pre>

			}
--	--	--	---

TC ID	Prošel test? ANO/NE	Očekávaný výsledek	Skutečný výsledek
18.	ANO	1. Stavový kód: 405 Method Not Allowed 2. Tělo odpovědi zobrazí odpovídající zprávu.	1. Stavový kód: 405 Method Not Allowed 2. Odpověď: { "timestamp": "2024-06-10T15:58:02.278+00:00", "status": 405, "error": "Method Not Allowed", "message": "", "path": "/api/v1/students/" }

BUG REPORT

Na základě provedených scénářů jsem objevila uvedené chyby aplikace:

TC ID: 2, 4, 5, 6, 7, 8, 10, 13, 14, 16