

FINÁLNÍ PROJEKT

č.1



Autor: Ing. Jaroslav Krcho
Datum: 31. 7. 2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	15
BUG REPORT	24

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:

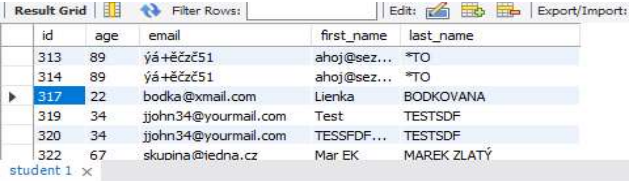

Nezapomeňte, že v IT se data musí někde uložit a poté získat. Proto ověřte, že data jsou správně uložena a získávána z databáze.

Nezapomeňte do testovacích scénářů uvést testovací data, očekávaný výsledek včetně těla odpovědi a stavových kódů.

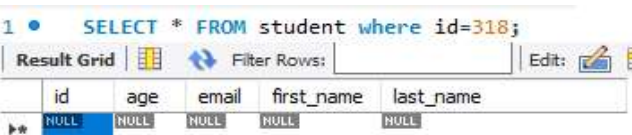
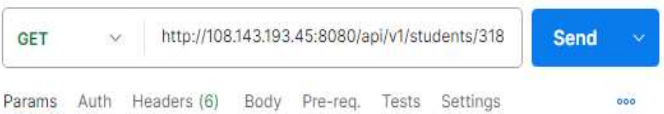
TESTOVACÍ SCÉNÁŘE

1. Method GET

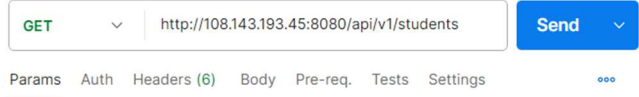
Test_01

Abstract:	
<i>Get data about existing student</i>	
Test data preparation:	Screen:
<i>In database student find an existing student. For example under id=317</i> Check: SELECT * FROM student where id=317; Answer: '317','22','bodka@xmail.com','Lienka','BODKOVA'	
Steps:	Screen:
<i>Find an existing student id=317</i> 1.) Select method GET and insert url: http://108.143.193.45:8080/api/v1/students/317 2.) Press Send button	
Expected result:	
1.) status code 200 OK 2.) Response: { id:317, firstName: "Lienka", lastName: "BODKOVA", email: " bodka@xmail.com". age: 22 }	

Test_02


Abstract:	
<i>Get data about non-existing student</i>	
Test data preparation:	Screen:
<i>In database student find an id that does not exist. For example id=318 does not contain any datas.</i> Check: SELECT * FROM student where id=318; Answer: 0 row(s) returned	
Steps:	Screen:
<i>Find the non-existing student id=318</i> 1.) Select method GET and insert url: http://108.143.193.45:8080/api/v1/students/318 2.) Press Send button	
Expected result:	
Status code 404 Not found	

Test 03

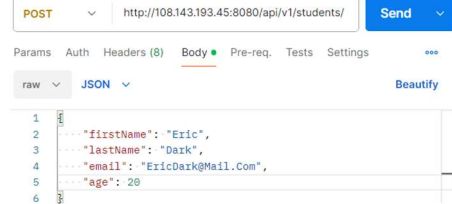
<u>Abstract:</u>	
<i>Get data without any parameter</i>	
<u>Steps:</u>	<u>Screen:</u>
1.) Select method GET and insert url: http://108.143.193.45:8080/api/v1/students 2.) Press Send button	
<u>Expected result:</u>	
Status code 400 Bad request	

2. Method POST


Test 04

Abstract:	
<i>Create a new student with the required attributes</i>	
Test data preparation:	
<p><i>New student data:</i></p> <p>First name: patrik Last name: hviezda Email: patrik.h@mail.com Age: 19</p>	
Steps:	Screen:
<ol style="list-style-type: none"> 1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type data of a new student 4.) Press Send button 	
Expected result:	
<ol style="list-style-type: none"> 1.) status code 201 created 2.) Response: { id:???, firstName: "patrik", lastName: "hviezda", email: "patrik.h@mail.com". age: 19 } 	<ol style="list-style-type: none"> 3. Database check: according assigned id.


Test 05

Abstract:	
<i>Create a new student, verify lowerCase() format</i>	
Test data preparation:	
<p><i>New student data:</i></p> <p>First name: Eric Last name: Dark Email: EricDark@Mail.Com Age: 20</p>	
Steps:	Screen:
<ol style="list-style-type: none"> 1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button 	
Expected result:	
<ol style="list-style-type: none"> 1.) Response: { id:???, firstName: "eric", lastName: "dark", email: "ericdark@mail.com". age: 20 } 	
2.) Database check: according assigned id	


Test_06

Abstract:	
Create a new student, verify invalid firstName, lastName, email and age	
Test data preparation:	
New student data: First name: Jo Last name: Fu Email: jofu@mail Age: 0	
Steps:	Screen:
1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button	
Expected result:	
status code 400 bad request 1.) "firstName must be between 3-50 characters" 2.) "lastName must be between 3-50 characters" 3.) "email must be in format <u>*@*.*</u> " 4.) "age must be between 1-150"	

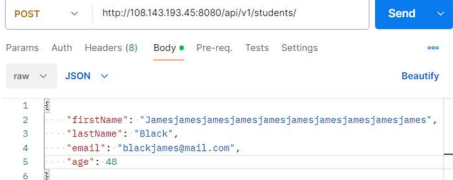
Test_07

Abstract:	
Create a new student, verify lower border value of firstName and age	
Test data preparation:	
New student data: First name: Bud Last name: Spencer Email: bud.spencer@mail.com Age: 1	
Steps:	Screen:
1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button	
Expected result:	
1.) Response: <pre>{ id:???, firstName: "bud", lastName: "spencer", email: "bud.spencer@mail.com". age: 1 }</pre> 2.) Database check: according assigned id	

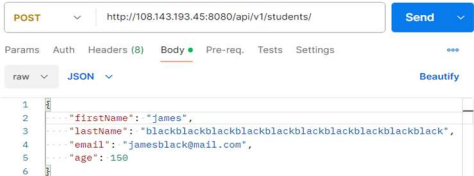
Test 08

Abstract:	
Create a new student, verify lower border value of lastName	
Test data preparation:	
New student data: First name: Marianna Last name: Tip Email: mariannatip40@mail.com Age: 40	
Steps:	Screen:
1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button	
Expected result:	
1.) Response: <pre>{ id:???, firstName: "marianna", lastName: "tip", email: "mariannatip40@mail.com". age: 40 }</pre> 2.) Database check: according assigned id	

Test 09

Abstract:	
Create a new student, verify upper border value of firstName	
Test data preparation:	
New student data: First name: (50characters) Jamesjamesjamesjamesjamesjamesjamesjamesjamesjames Last name: Black Email: blackjames@mail.com Age: 48	
Steps:	Screen:
1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button	
Expected result:	
1.) Response: <pre>{ id:???, firstName: "jamesjamesjamesjamesjamesjamesjamesjamesjamesjames", lastName: "black", email: "blackjames@mail.com". age: 48 }</pre> 2.) Database check: according assigned id	


Test_10

<p><u>Abstract:</u></p> <p>Create a new student, verify upper border value of lastName and age</p>	
<p><u>Test data preparation:</u></p> <p>New student data:</p> <p>First name: James</p> <p>Last name: (50 characters)</p> <p>blackblackblackblackblackblackblackblackblackblackblack</p> <p>Email: jamesblack@mail.com</p> <p>Age: 150</p>	
<p><u>Steps:</u></p> <ol style="list-style-type: none"> 1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button 	<p><u>Screen:</u></p> 
<p><u>Expected result:</u></p>	
<ol style="list-style-type: none"> 1.) Response: <pre>{ id:???, firstName: "james", lastName: "blackblackblackblackblackblackblackblackblackblack", email: "jamesblack@mail.com". age: 150 }</pre> 2.) Database check: according assigned id 	


Test 11

<u>Abstract:</u>	
Create a new student, verify invalid values of firstName, lastName and age (all over upper border values)	
<u>Test data preparation:</u>	
New student data: First name: (55 characters) jamesjamesjamesjamesjamesjamesjamesjamesjamesjamesjamesjames Last name: (55 characters) blackblackblackblackblackblackblackblackblackblackblackblack Email: jamesblack1@mail.com Age: 180	
<u>Steps:</u>	<u>Screen:</u>
1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button	 <pre>1 { 2 "firstName": "jamesjamesjamesjamesjamesjamesjamesjamesjamesjamesjamesjames", 3 "lastName": "blackblackblackblackblackblackblackblackblackblackblackblack", 4 "email": "jamesblack1@mail.com", 5 "age": 180 6 }</pre>
<u>Expected result:</u>	
Status code 400 bad request 1.) "firstName must be between 3-50 characters" 2.) "lastName must be between 3-50 characters" 3.) "age must be between 1-150"	


Test 12

Abstract:	
Create a new student, verify invalid values of firstName (nothing input) and lastName (input incorrect characters)	
Test data preparation:	
<p>New student data:</p> <p>First name:</p> <p>Last name: St@ne{</p> <p>Email: john.stone@mail.com</p> <p>Age: 46</p>	
Steps:	Screen:
<ol style="list-style-type: none"> 1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button 	
Expected result:	
<p>Status code 400 bad request</p> <ol style="list-style-type: none"> 1.) firstName is mandatory 2.) lastName contains invalid characters 	


Test 13

Abstract:	
Create a new student, verify invalid values of lastName (nothing input) and email (nothing input)	
Test data preparation:	
<p>New student data:</p> <p>First name: jessica</p> <p>Last name:</p> <p>Email:</p> <p>Age: 39</p>	
Steps:	Screen:
<ol style="list-style-type: none"> 1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button 	
Expected result:	
<p>Status code 400 bad request</p> <ol style="list-style-type: none"> 1.) lastName is mandatory 2.) email is mandatory 	


Test 14

Abstract:	
Create a new student, verify invalid value of age (input alphabet characters)	
Test data preparation:	
New student data: First name: kevin Last name: sting Email: kevin.sting@mail.com Age: abc	
Steps:	Screen:
1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button	
Expected result:	
Status code 400 bad request Age contains invalid characters	


Test 15

Abstract:	
Create a new student, verify invalid value of age (nothing input)	
Test data preparation:	
New student data: First name: jennifer Last name: gold Email: jennifergold25@xmail.com Age:	
Steps:	Screen:
1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button	
Expected result:	
Status code 400 bad request Age is mandatory	


Test 16

Abstract:	
Create a new student, the attribute <i>firstName</i> is missing	
Test data preparation:	
New student data:	
Last name: Smith Email: johnsmith@post.com Age: 28	
Steps:	Screen:
1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button	
Expected result:	
Status code 400 bad request firstName is mandatory	

Test 17

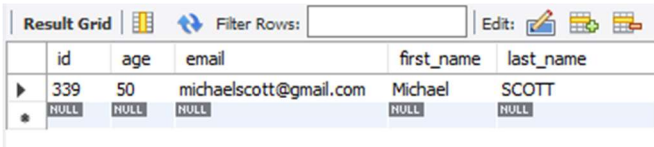
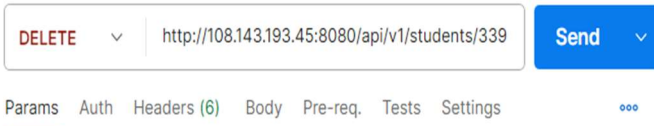
Abstract:	
Create a new student, the attribute <i>lastName</i> is missing	
Test data preparation:	
New student data:	
First name: John Email: johnsmith@post.com Age: 28	
Steps:	Screen:
1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button	
Expected result:	
Status code 400 bad request firstName is mandatory	

Test 18

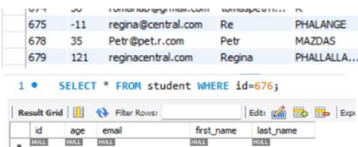
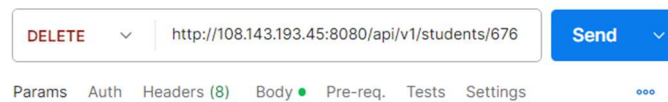
<u>Abstract:</u>	
<i>Create a new student, the attribute email is missing</i>	
<u>Test data preparation:</u>	
New student data: First name: John Last name: Smith Age: 28	
<u>Steps:</u>	
1.) Select method POST and insert url: http://108.143.193.45:8080/api/v1/students/ 2.) Click to Body, choose raw and JSON format. 3.) type the prepared data of a new student 4.) Press Send button	<u>Screen:</u> 
<u>Expected result:</u>	
Status code 400 bad request email is mandatory	

3. Method DELETE

Test 19

<u>Abstract:</u>																			
Delete data of existing student																			
<u>Test data preparation:</u>	<u>Screen:</u>																		
<p>Find an existing student id=339</p> <pre>SELECT * FROM student where id=339; '339','50','michaelscott@gmail.com ','Michael','SCOTT'</pre>	 <table><thead><tr><th></th><th>id</th><th>age</th><th>email</th><th>first_name</th><th>last_name</th></tr></thead><tbody><tr><td>▶</td><td>339</td><td>50</td><td>michaelscott@gmail.com</td><td>Michael</td><td>SCOTT</td></tr><tr><td>*</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></tbody></table>		id	age	email	first_name	last_name	▶	339	50	michaelscott@gmail.com	Michael	SCOTT	*	NULL	NULL	NULL	NULL	NULL
	id	age	email	first_name	last_name														
▶	339	50	michaelscott@gmail.com	Michael	SCOTT														
*	NULL	NULL	NULL	NULL	NULL														
<u>Steps:</u>	<u>Screen:</u>																		
<p>1.) Select method DELETE and insert url: http://108.143.193.45:8080/api/v1/students/339</p> <p>2.) Press Send button</p>	 <p>DELETE http://108.143.193.45:8080/api/v1/students/339 Send</p> <p>Params Auth Headers (6) Body Pre-req. Tests Settings ...</p>																		
<u>Expected result:</u>																			
<p>1.) status code 200 OK</p> <p>2.) Database check: All data about particular student data are REMOVED</p>																			

Test 20

<u>Abstract:</u>	
Delete data of non-existing student	
<u>Test data preparation:</u>	<u>Screens:</u>
<p>Find an id that does not exist (id=676)</p> <p>SELECT * FROM student where id=676;</p> <p>0 row(s) returned</p>	
<u>Steps:</u>	<u>Screen:</u>
<p>1.) Select method DELETE and insert url: http://108.143.193.45:8080/api/v1/students/676</p> <p>2.) Press Send button</p>	
<u>Expected result:</u>	
<p>1.) status code 404</p>	

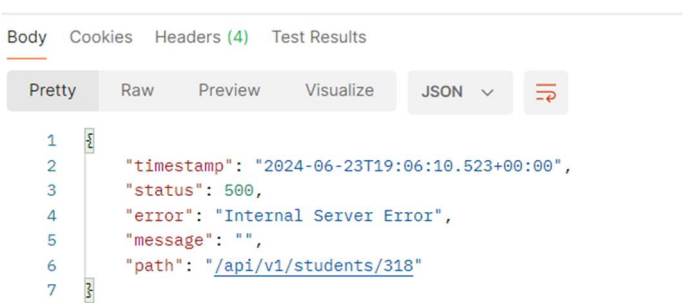
EXEKUCE TESTŮ

Testovací scénáře jsem provedl, příkládám výsledky testů.

Test_01

Actual result:	Documentation:
<div>Status code: 200 OK</div> <div><pre>{ "id": 317, "firstName": "Lienka", "lastName": "BODKOVANA", "email": "bodka@xmail.com", "age": 22 }</pre></div>	<div></div>
Test result:	The test passed

Test_02

Actual result:	Documentation:
<div>Status code: 500 Internal Server Error</div> <div><pre>{ "timestamp": " 2024-06-23T19:06:10.523+00:00", "status": 500, "error": "Internal Server Error", "message": "", "path": "/api/v1/students/318" }</pre></div>	<div></div>
Test result:	The test failed

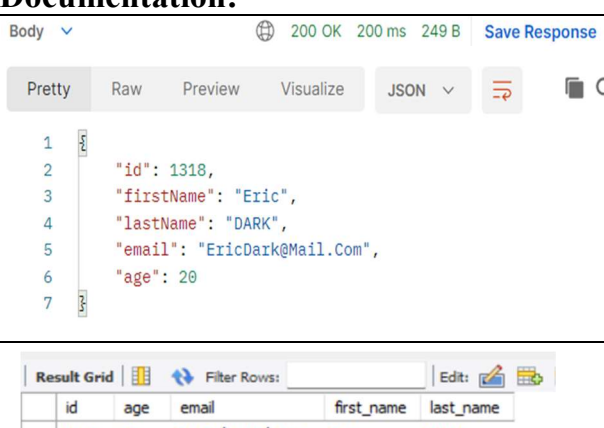
Test_03

Actual result:	Documentation:
<div>Status code: 200 OK</div> <div>Showned all data about all students</div>	<div></div>
Test result:	The test failed

Test_04

Actual result:	Documentation:
Status code: 200 OK <pre>{ "id": 1202, "firstName": "patrik", "lastName": "HVIEZDA", "email": "patrik.h@mail.com", "age": 19 }</pre>	
Database check	
Test result:	The test failed

Test_05

Actual result:	Documentation:
Status code: 200 OK <pre>{ "id": 1318, "firstName": "Eric", "lastName": "DARK", "email": "EricDark@Mail.Com", "age": 20 }</pre>	
Database check	
Notice:	1.) Wrong status code 200 OK is known bug – reported in Test_04 2.) Capital letters in lastName is known bug – reported in Test_04
Test result:	The test failed

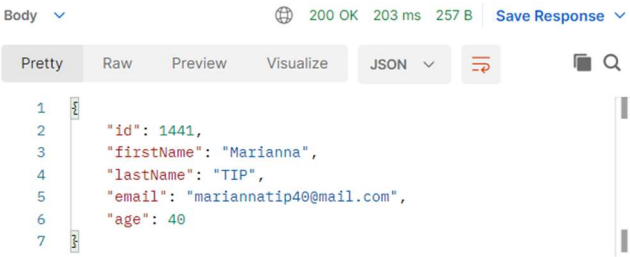
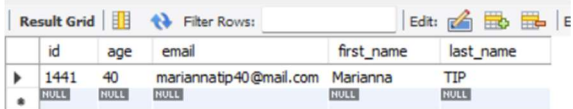
Test_06

Actual result:	Documentation:																		
<p>The student was created.</p> <p>Status code: 200 OK</p> <pre>{ "id": 1319, "firstName": "Jo", "lastName": "FU", "email": "jofu@mail", "age": 0 }</pre>	<div>Body 200 OK 195 ms 236 B Save Response</div> <div><div>PrettyRawPreviewVisualizeJSON </div><div></div></div> <pre>1 { 2 "id": 1319, 3 "firstName": "Jo", 4 "lastName": "FU", 5 "email": "jofu@mail", 6 "age": 0 7 }</pre>																		
Database check	<div><div>Result Grid </div><div>Filter Rows: <input type="text"/></div><div>Edit: </div></div> <table><tr><th></th><th>id</th><th>age</th><th>email</th><th>first_name</th><th>last_name</th></tr><tr><td>▶</td><td>1319</td><td>0</td><td>jofu@mail</td><td>Jo</td><td>FU</td></tr><tr><td>✱</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table>		id	age	email	first_name	last_name	▶	1319	0	jofu@mail	Jo	FU	✱	NULL	NULL	NULL	NULL	NULL
	id	age	email	first_name	last_name														
▶	1319	0	jofu@mail	Jo	FU														
✱	NULL	NULL	NULL	NULL	NULL														
Notice:	Capital letters in lastName is known bug – reported in Test_04																		
Test result:	The test failed																		

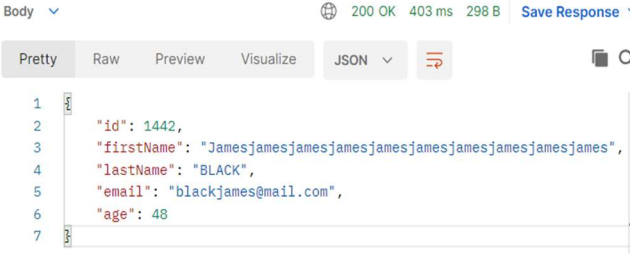
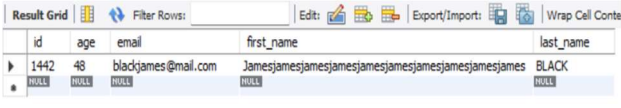
Test_07

Actual result:	Documentation:															
<p>The student was created.</p> <p>Status code: 200 OK</p> <pre>{ "id": 1432, "firstName": "Bud", "lastName": "SPENCER", "email": "bud.spencer@mail.com", "age": 1 }</pre>	<div><div>Body </div><div>200 OK 171 ms 253 B Save Response</div><div><div>PrettyRawPreviewVisualizeJSON </div><div></div><div></div></div><pre>1 { 2 "id": 1432, 3 "firstName": "Bud", 4 "lastName": "SPENCER", 5 "email": "bud.spencer@mail.com", 6 "age": 1 7 }</pre></div>															
Database check	<div><div>Result Grid </div><div>Filter Rows: <input type="text"/></div><div>Edit: </div><div></div><div></div><div></div></div> <table><tr><th>id</th><th>age</th><th>email</th><th>first_name</th><th>last_name</th></tr><tr><td>1432</td><td>1</td><td>bud.spencer@mail.com</td><td>Bud</td><td>SPENCER</td></tr><tr><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table>	id	age	email	first_name	last_name	1432	1	bud.spencer@mail.com	Bud	SPENCER	NULL	NULL	NULL	NULL	NULL
id	age	email	first_name	last_name												
1432	1	bud.spencer@mail.com	Bud	SPENCER												
NULL	NULL	NULL	NULL	NULL												
Notice:	<div><div>1.) Wrong status code 200 OK is known bug – reported in Test_04</div><div>2.) Capital letters in lastName is known bug – reported in Test_04</div></div>															
Test result:	The test passed															

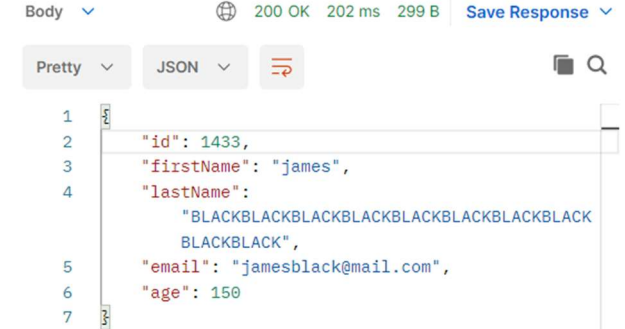
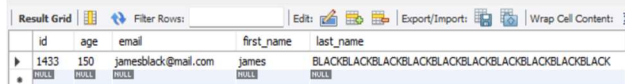
Test_08

Actual result:	Documentation:
<p>The student was created. Status code: 200 OK</p> <pre>{ "id": 1441, "firstName": "Marianna", "lastName": "TIP", "email": "mariannatip40@mail.com", "age": 40 }</pre>	
Database check	
Notice:	<ol style="list-style-type: none"> 1.) Wrong status code 200 OK is known bug – reported in Test_04 2.) Capital letters in lastName is known bug – reported in Test_04
Test result:	The test passed

Test_09

Actual result:	Documentation:
<p>The student was created. Status code: 200 OK</p> <pre>{ "id": 1442, "firstName": "Jamesjamesjamesjames jamesjamesjamesjamesjamesjames", "lastName": "BLACK", "email": "blackjames@mail.com", "age": 48 }</pre>	
Database check	
Notice:	<ol style="list-style-type: none"> 1.) Wrong status code 200 OK is known bug – reported in Test_04 2.) Capital letters in lastName is known bug – reported in Test_04
Test result:	The test passed

Test_10

<p>Actual result:</p> <p>The student was created. Status code: 200 OK</p> <pre>{ "id": 1433, "firstName": "james", "lastName": "BLACKBLACKBLACKBLACKBLACKBLACKBLACKBLACKBLACKBLACKBLACKBLACK", "email": "jamesblack@mail.com", "age": 150 }</pre>	<p>Documentation:</p> 
<p>Database check</p>	
<p>Notice:</p>	<p>1.) Wrong status code 200 OK is known bug – reported in Test_04 2.) Capital letters in lastName is known bug – reported in Test_04</p>
<p>Test result:</p>	<p>The test passed</p>

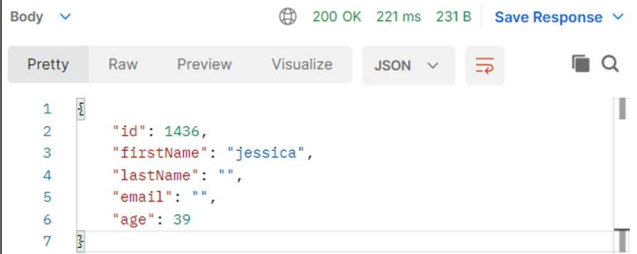
Test_11

<p>Actual result:</p> <p>The student was created.</p> <p>Status code: 200 OK</p> <pre>{ "id": 1434, "firstName": "jamesjamesjamesjamesjames jamesjamesjamesjamesjamesjames", "lastName": "BLACKBLACKBLACKBLACKBLACKB LACKBLACKBLACKBLACKBLACKBLACK", "email": "jamesblack1@mail.com", "age": 180 }</pre>	<p>Documentation:</p> 
<p>Database check</p>	
<p>Notice:</p>	<p>Capital letters in lastName is known bug – reported in Test_04</p>
<p>Test result:</p>	<p>The test failed</p>

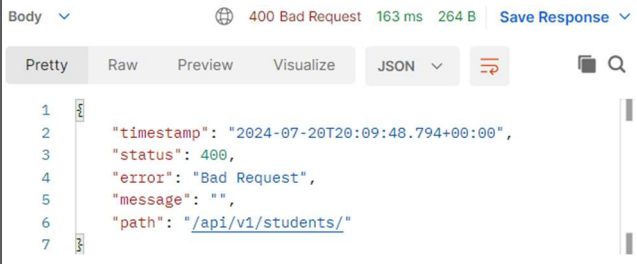
Test_12

Actual result:	Documentation:
<p>The student was created. Status code: 200 OK</p> <pre>{ "id": 1435, "firstName": "", "lastName": "ST@NE{", "email": "john.stone@mail.com", "age": 46 }</pre>	
Database check	
Test result:	The test failed

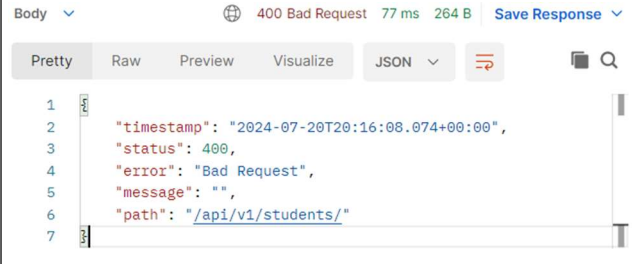
Test_13

Actual result:	Documentation:
<p>The student was created. Status code: 200 OK</p> <pre>{ "id": 1436, "firstName": "jessica", "lastName": "", "email": "", "age": 39 }</pre>	
Database check	
Test result:	The test failed

Test_14

Actual result:	Documentation:
Status code: 400 Bad Request <pre>{ "timestamp": "2024-07-20T20:09:48.794+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/" }</pre>	
Test result:	The test passed

Test_15

Actual result:	Documentation:
Status code: 400 Bad Request <pre>{ "timestamp": "2024-07-20T20:16:08.074+00:00", "status": 400, "error": "Bad Request", "message": "", "path": "/api/v1/students/" }</pre>	
Test result:	The test passed

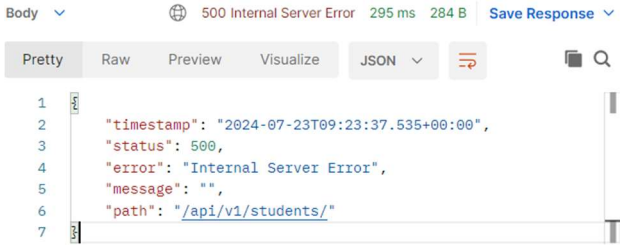
Test_16

Actual result:	Documentation:
Status code: 500 Internal Server Error <pre>{ "timestamp": "2024-07-23T09:06:47.773+00:00", "status": 500, "error": "Internal Server Error", "message": "", "path": "/api/v1/students/" }</pre>	
Test result:	The test failed

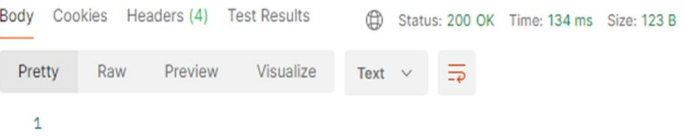
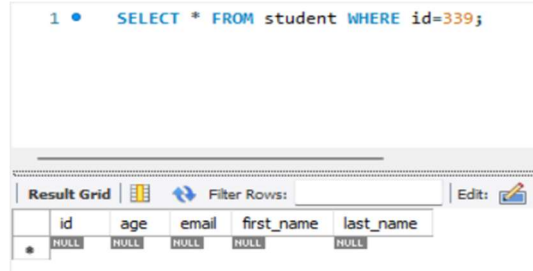
Test_17

Actual result:	Documentation:
Status code: 500 Internal Server Error <pre>{ "timestamp": "2024-07-23T09:18:28.195+00:00", "status": 500, "error": "Internal Server Error", "message": "", "path": "/api/v1/students/" }</pre>	
Test result:	The test failed

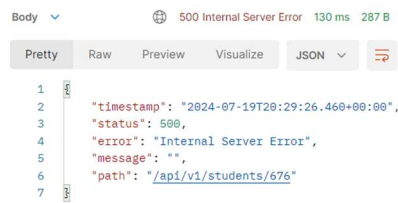
Test_18

Actual result:	Documentation:
Status code: 500 Internal Server Error <pre>{ "timestamp": "2024-07-23T09:23:37.535+00:00", "status": 500, "error": "Internal Server Error", "message": "", "path": "/api/v1/students/" }</pre>	
Test result:	The test failed

Test_19

Actual result:	Documentation:
The student was deleted. Status code 200 OK	
All data about particular student data are REMOVED	
Test result:	The test passed

Test_20

Actual result:	Documentation:
Status code: 500 Internal Server Error	 <pre>Body 500 Internal Server Error 130 ms 287 B Pretty Raw Preview Visualize JSON 1 { 2 "timestamp": "2024-07-19T20:29:26.460+00:00", 3 "status": 500, 4 "error": "Internal Server Error", 5 "message": "", 6 "path": "/api/v1/students/676" 7 }</pre>
Test result:	The test failed

BUG REPORT

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

- Test_02: wrong status code – 500 Internal Server Error instead of 404 Not found
- Test_03: 1. wrong status code – 200 OK instead of 400 Bad request
2. Showed all data about all students.
- Test_04: 1. wrong status code after creating a new student – 200 OK instead of 201 created
2. the last name is saved with capitals letters
- Test_05: The student was created but the lowerCase() format is not followed
- Test_06: 1. wrong status code – 200 OK instead of 400 Bad request
2. The student was created in spite of invalid data of firstName, lastName, email and age.
- Test_11: 1. wrong status code – 200 OK instead of 400 Bad request
2. The student was created in spite of invalid data of firstName, lastName, email and age (all values over upper border value)
- Test_12: 1. wrong status code – 200 OK instead of 400 Bad request
2. The student was created in spite of invalid data of firstName (nothing input) and lastName (incorrect characters).
- Test_13: 1. wrong status code – 200 OK instead of 400 Bad request
2. The student was created in spite of invalid data of lastName (nothing input) and email (nothing input).
- Test_14: No message about invalid characters was showed
- Test_15: No message about the mandatory of age was showed
- Test_16: Wrong status code – 500 Internal Server Error instead of code 400 Bad request
- Test_17: Wrong status code – 500 Internal Server Error instead of code 400 Bad request
- Test_18: Wrong status code – 500 Internal Server Error instead of code 400 Bad request
- Test_20: wrong status code – 500 Internal Server Error instead of code 404 Not found