

FINÁLNÍ PROJEKT

č.1



Autor: Martin Vácha
Datum: 28.04.2024

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	2
BUG REPORT	2

1 ZADÁNÍ

Čílem 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..

1.1 Přístupové údaje:

Databáze	database: qa_demo Host: aws.connect.psdb.cloud Username: q524i16o56a3hjktx53r Password: pscale_pw_JvkHoSxulxh8GLHYEDCzCvgO qyAgkU7zfNspWwhMoCX
REST-API	http://108.143.193.45:8080/api/v1/students/

1.2 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.

2 TESTOVACÍ SCÉNÁŘE

Na základě uvedených testovacích scénářů jsem ověřil(a) funkčnost aplikace.

2.1 Test metoda GET

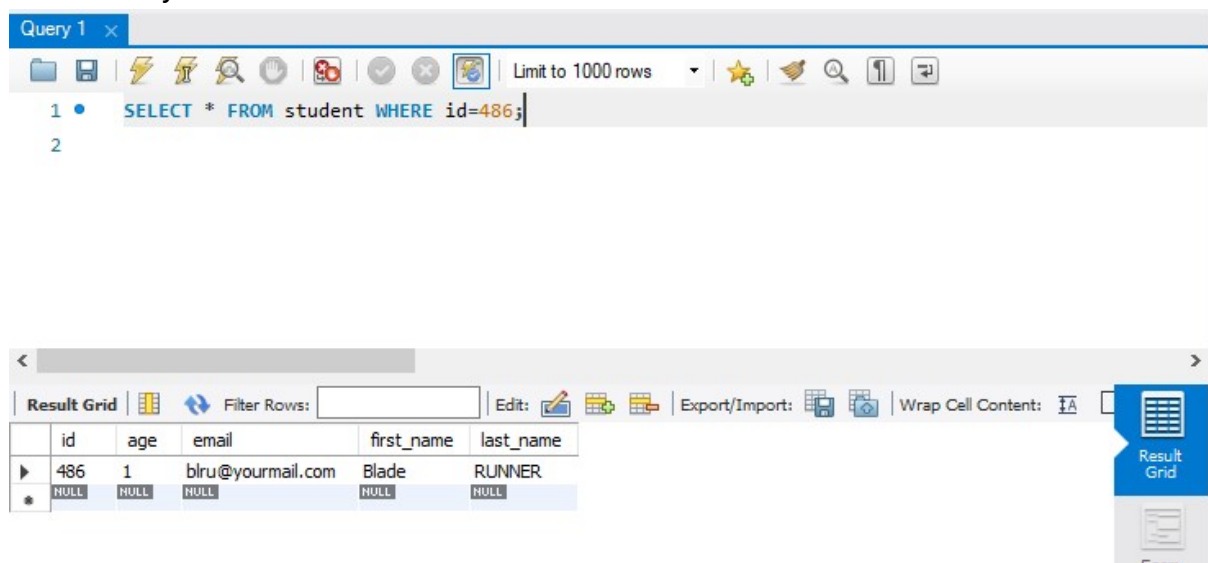
Zadáno:

Vyhledán existující student 500

Do Postman:

<http://108.143.193.45:8080/api/v1/students/486>

Kontrola MySQL Workbench



<http://108.143.193.45:8080/api/v1/students/486>

Očekávaný výsledek:

```
{
  "id": 486,
  "firstName": "Blade",
  "lastName": "RUNNER",
  "email": "blru@yourmail.com",
  "age": 1
}
```

Reálný výsledek:

```
{
  "id": 486,
  "firstName": "Blade",
  "lastName": "RUNNER",
  "email": "blru@yourmail.com",
  "age": 1
}
```

Výsledek:

Test v pořádku (očekávaný = reálný výsledek)

2.2 Test metoda GET

Zadáno:

Vyhledán „neexistující“ student 500

Do Postman:

<http://108.143.193.45:8080/api/v1/students/500>

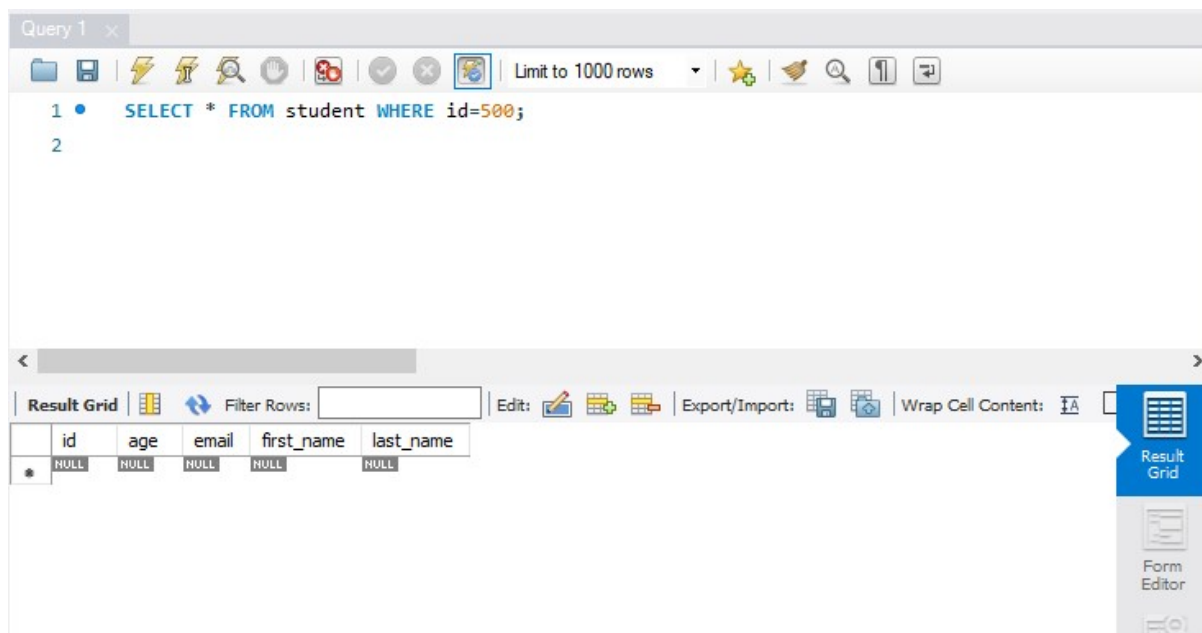
Postman interface showing a GET request to `http://108.143.193.45:8080/api/v1/students/500`. The request body is a JSON object:

```
1 {
2   "firstName": "Blade",
3   "lastName": "Runner",
4   "age": 1,
5   "email": "blru@yourmail.com"
6 }
```

The response is a 500 Internal Server Error. The response body is a JSON object:

```
1 {
2   "timestamp": "2024-04-28T14:27:50.841+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "message": "",
6   "path": "/api/v1/students/500"
7 }
```

Kontrola MySQL Workbench:



Očekávaný výsledek:

Error

Reálný výsledek:

Error 500 - Internal server error - no more specific message is suitable

```
{
  "timestamp": "2024-04-28T14:11:32.939+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "message": "",
  "path": "/api/v1/students/500"
}
```

Výsledek:

Test v pořádku (očekávaný = reálný výsledek)

Výsledek by mohl být více zpecifický – oznámit – „Student XY (500) neexistuje.“

2.3 Test metoda POST

Zadáno:

Nový student:

```
{
  "firstName": "Ivan",
  "lastName": "Ivanovic",
  "age": -10,
  "email": "blru"
}
```

Do Postman:

http://108.143.193.45:8080/api/v1/students/

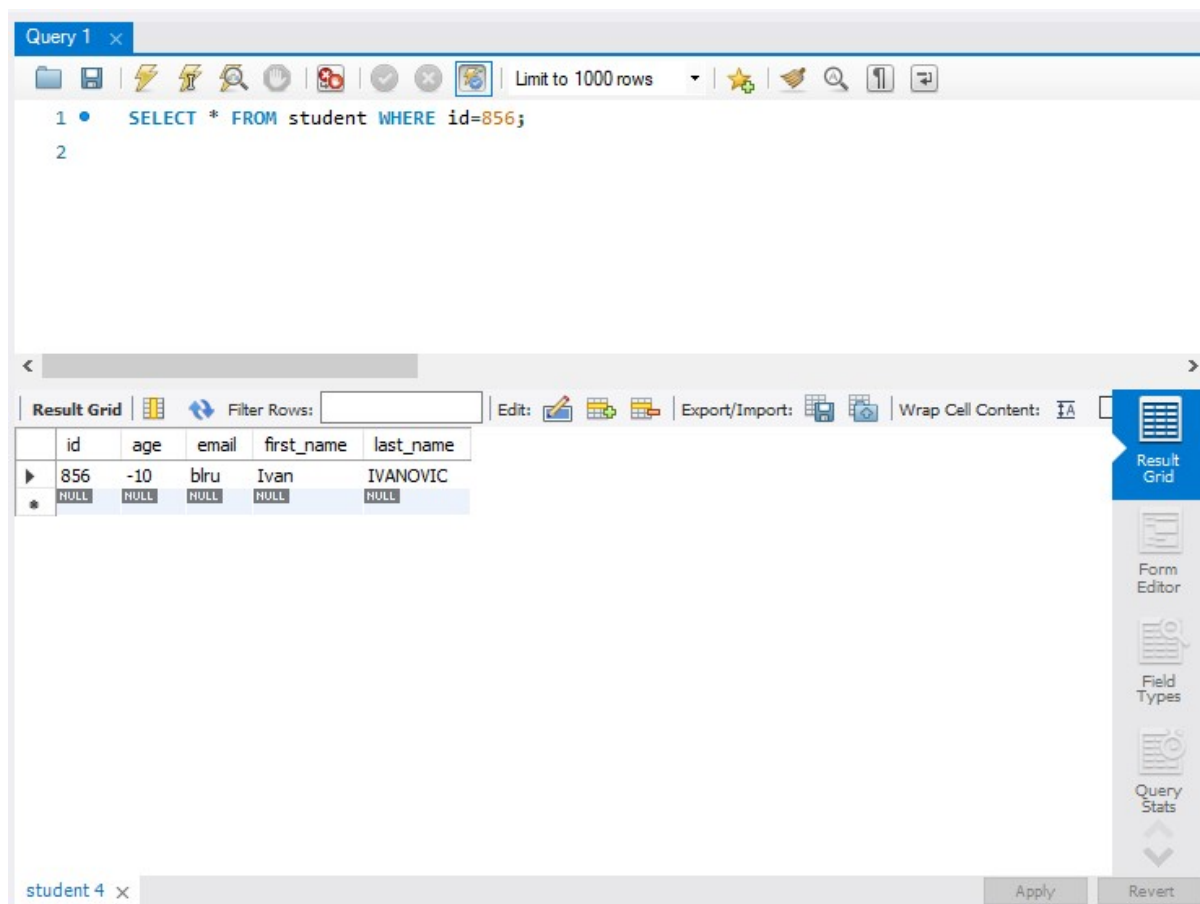
The image shows the Postman interface for a POST request. The URL is `http://108.143.193.45:8080/api/v1/students/`. The request method is **POST**. The body is set to **raw** with the following JSON content:

```
1 {
2   "firstName": "Ivan",
3   "lastName": "Ivanovic",
4   "age": -10,
5   "email": "blru"
6 }
```

The response status is **200 OK** with a response time of **178 ms** and a body size of **240 B**. The response body is displayed in the **Body** tab, showing the following JSON object:

```
1 {
2   "id": 856,
3   "firstName": "Ivan",
4   "lastName": "IVANOVIC",
5   "email": "blru",
6   "age": -10
7 }
```

Kontrola MySQL Workbench:



Očekávaný výsledek:

Error

- Nelze zadat zápornou hodnotu (-10) do pole "age"
- Neplatný email – email musí obsahovat @ (př.: engeto@engeto.cz)

Reálný výsledek:

```
{
  "id": 856,
  "firstName": "Ivan",
  "lastName": "IVANOVIC",
  "email": "blru",
  "age": -10
}
```

Výsledek:

Test je špatně (očekávaný ≠ reálný výsledek)

Systém dovolil vytvoření nereálného emailu.

Systém dovolil zadat zápornou hodnotu pro pole "age" (věk).

2.4 Test metoda POST

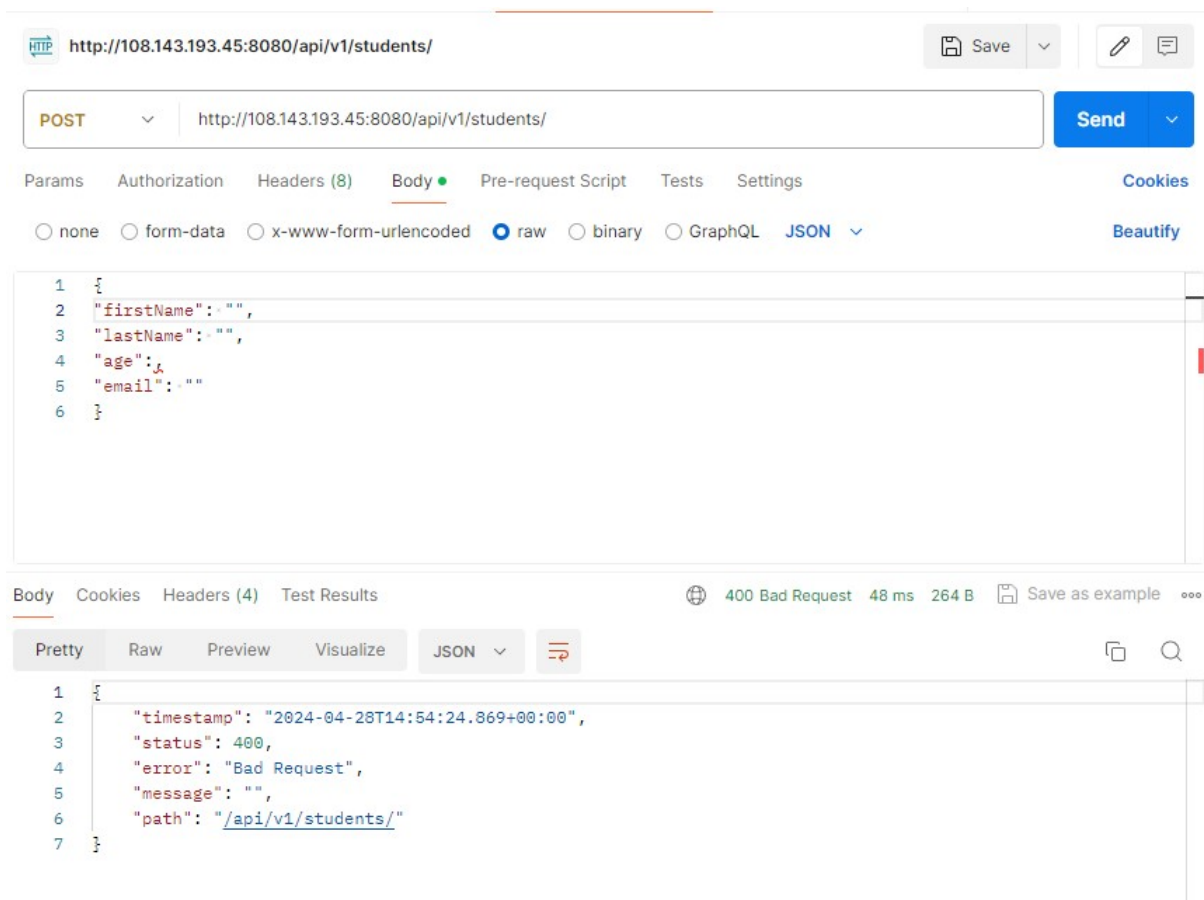
Zadáno:

Nový student:

```
{
  "firstName": "",
  "lastName": "",
  "age": ,
  "email": ""
}
```

Do Postman:

http://108.143.193.45:8080/api/v1/students/



Kontrola MySQL Workbench:

Očekávaný výsledek:

Error

Musí být zadán

```
"firstName"
"lastName"
"age"
"email"
```

Reálný výsledek:

Error 400 - Bad request

```
{
  "timestamp": "2024-04-28T14:54:24.869+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "",
  "path": "/api/v1/students/"
}
```

Výsledek:

Test v pořádku (očekávaný = reálný výsledek)

Error více specifický – chybějící údaje - "firstName","lastName","age","email"

2.5 Test metoda POST

Zadáno:

Nový student:

```
{  
  "firstName": "Borec",  
  "lastName": "",  
  "age": 256,  
  "email": "borec1@borec.com"  
}
```

Do Postman:

<http://108.143.193.45:8080/api/v1/students/>

HTTP <http://108.143.193.45:8080/api/v1/students/> Save

POST <http://108.143.193.45:8080/api/v1/students/> Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

```
1 {
2   "firstName": "Borec",
3   "lastName": "",
4   "age": 256,
5   "email": "borec1@borec.com"
6 }
```

Body Cookies Headers (5) Test Results 200 OK 190 ms 245 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 857,
3   "firstName": "Borec",
4   "lastName": "",
5   "email": "borec1@borec.com",
6   "age": 256
7 }
```

Kontrola MySQL Workbench:

Query 1

Limit to 1000 rows

```
1 • SELECT * FROM student WHERE id=857;
2
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	id	age	email	first_name	last_name
▶	857	256	borec1@borec.com	Borec	
•	NULL	NULL	NULL	NULL	NULL

Result Grid Form Editor Field Types Query Stats

Očekávaný výsledek:

Error

- Chybí hodnota pole `"lastName"`
- Nereálná hodnota `"age"` ,

Reálný výsledek:

```
{  
  "id": 857,  
  "firstName": "Borec",  
  "lastName": "",  
  "email": "borec1@borec.com",  
  "age": 256  
}
```

Výsledek:

Test je špatně (očekávaný \neq reálný výsledek)

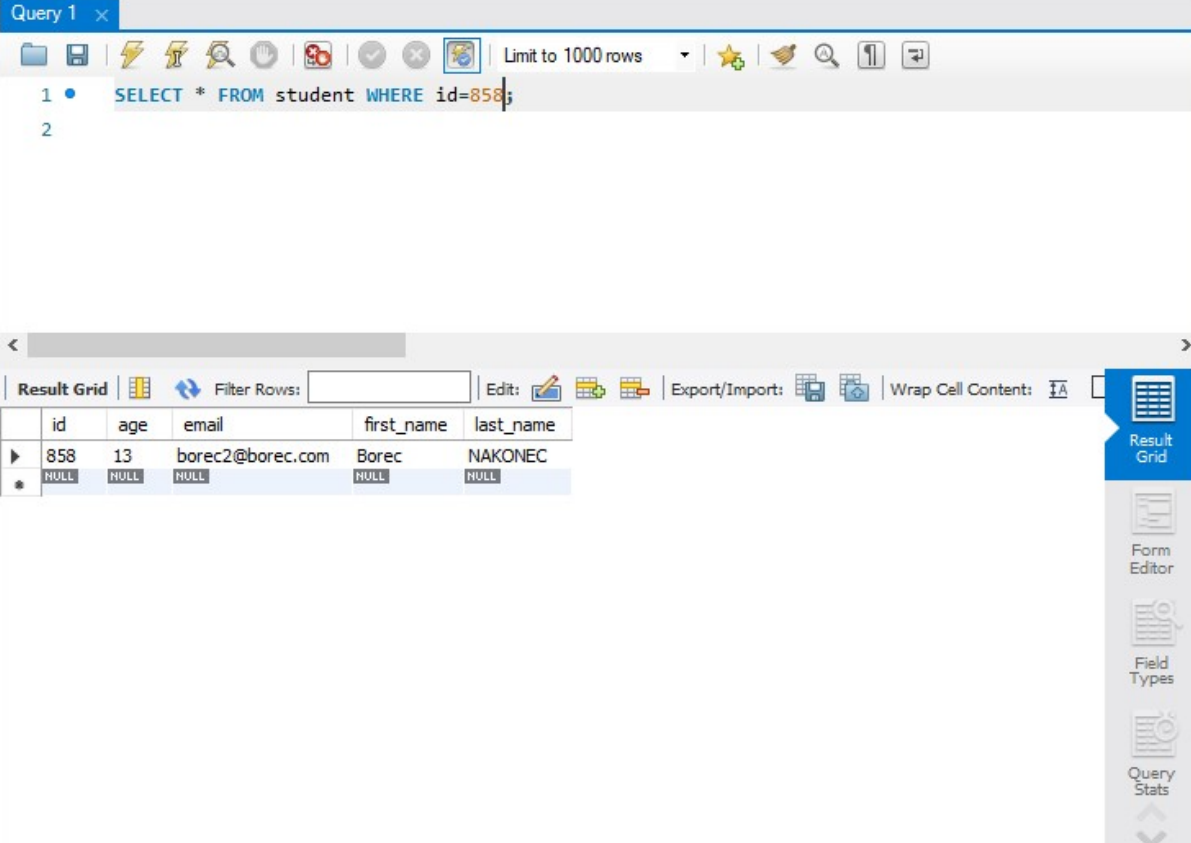
Systém dovolil vytvoření studenta bez zadání jednoho z polí (`"lastName"`)

Systém dovolil vytvoření studenta se zadáním nereálné hodnoty v poli `"age"` – mělo by být nastaveno omezení

2.6 Test metoda DELETE

Zadáno:

Smazání studenta 858:



The screenshot shows a database query tool interface. At the top, a query editor displays the SQL statement: `SELECT * FROM student WHERE id=858;`. Below the editor, a toolbar includes icons for saving, running, and other database operations, along with a 'Limit to 1000 rows' dropdown. The bottom section shows the 'Result Grid' with a table of data. The table has columns for 'id', 'age', 'email', 'first_name', and 'last_name'. The first row contains the data for student ID 858: Borec, NAKONEC. Below this row, there are two rows of 'NULL' values, indicating that the student with ID 858 was successfully deleted. On the right side of the interface, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'.

id	age	email	first_name	last_name
858	13	borec2@borec.com	Borec	NAKONEC
NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL

Do Postman:

<http://108.143.193.45:8080/api/v1/students/858>

Overview | DEL http://108.143.193.45:8080 | DEL http://108.143.193.45:8080 | + | No environment

HTTP http://108.143.193.45:8080/api/v1/students/858 | Save | Edit | Comments

DELETE | http://108.143.193.45:8080/api/v1/students/858 | Send

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings | Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL | JSON | Beautify

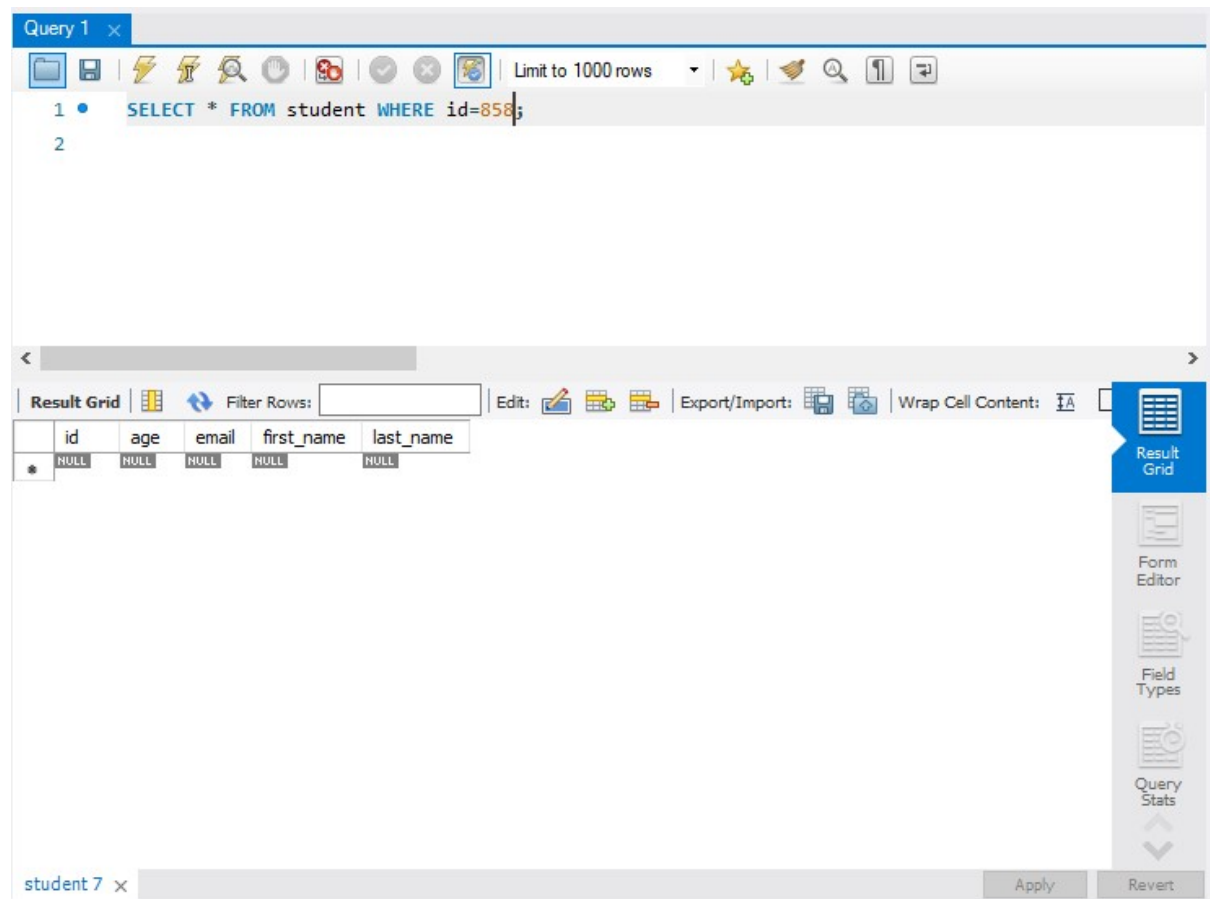
```
1 {
2   "firstName": "Borec",
3   "lastName": "Nakonec",
4   "age": 13,
5   "email": "borec2@borec.com"
6 }
```

Body | Cookies | Headers (4) | Test Results | 200 OK 120 ms 123 B | Save as example

Pretty | Raw | Preview | Visualize | Text | Copy | Search

1

Kontrola MySQL Workbench:



Očekávaný výsledek:
OK – Status 200

Reálný výsledek:
200 OK

Výsledek:

Test v pořádku (očekávaný = reálný výsledek)
Student byl smazán

2.7 Test metoda DELETE

Zadáno:

Smazání již smazaného studenta 858:

The screenshot shows a database query editor interface. At the top, there's a tab labeled "Query 1". Below it is a toolbar with various icons for file operations, execution, and settings. The main area contains a SQL query: `SELECT * FROM student WHERE id=858;`. Below the query, there's a "Result Grid" section with a table showing columns: id, age, email, first_name, last_name. The first row of data shows all values as NULL. On the right side, there's a vertical toolbar with icons for "Result Grid", "Form Editor", "Field Types", and "Query Stats". At the bottom, there's a status bar with "student 7" and buttons for "Apply" and "Revert".

id	age	email	first_name	last_name
NULL	NULL	NULL	NULL	NULL

Do Postman:

<http://108.143.193.45:8080/api/v1/students/858>

Overview x DEL http://108.143.193.45:8080 DEL http://108.143.193.45:8080 + No environment

HTTP http://108.143.193.45:8080/api/v1/students/858 Save

DELETE http://108.143.193.45:8080/api/v1/students/858 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON Beautify

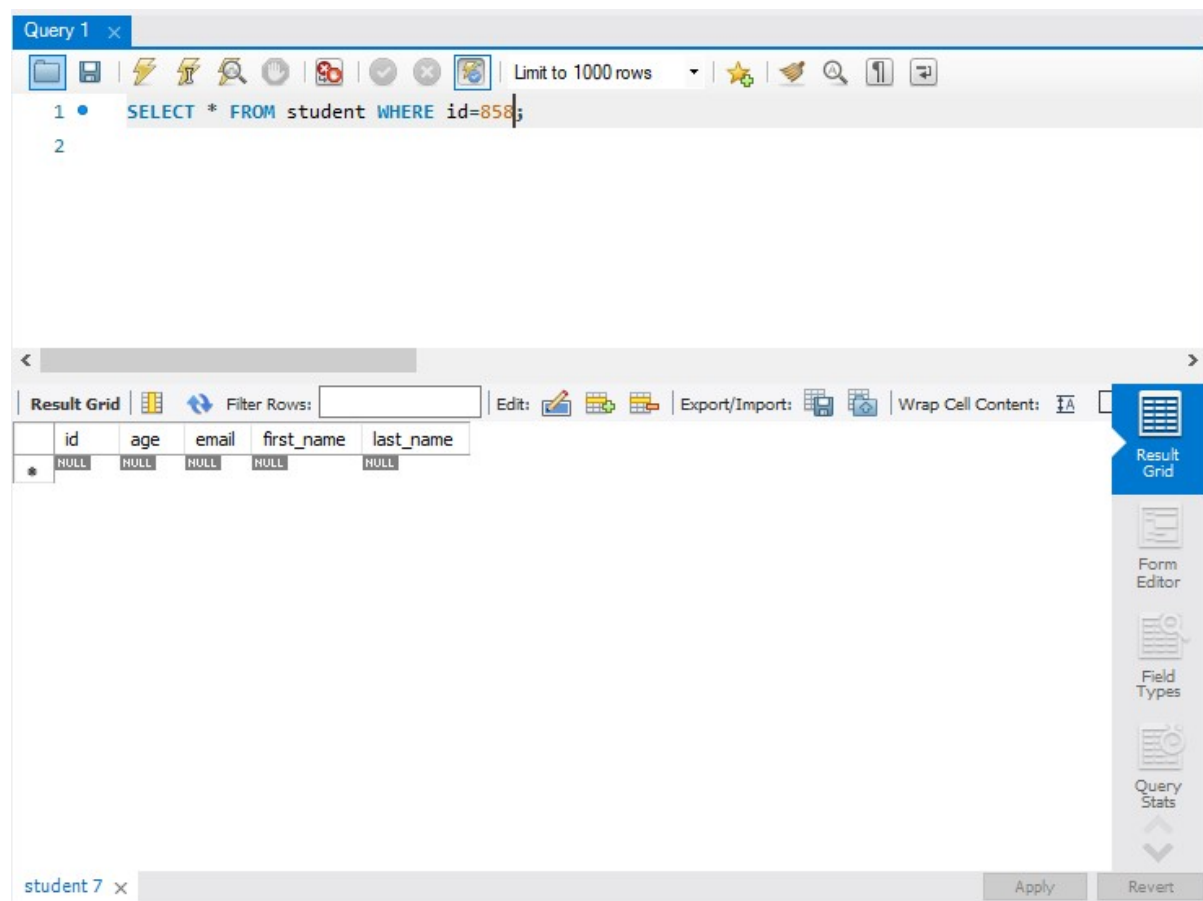
```
1 {
2   "firstName": "Kamil",
3   "lastName": "Ondra",
4   "age": 12,
5   "email": "borec3@borec.com"
6 }
```

Body Cookies Headers (4) Test Results 500 Internal Server Error 154 ms 287 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2024-04-28T15:23:44.858+00:00",
3   "status": 500,
4   "error": "Internal Server Error",
5   "message": "",
6   "path": "/api/v1/students/858"
7 }
```

Kontrola MySQL Workbench:



Očekávaný výsledek:

Error

Reálný výsledek:

500 – Internal server error

```
{
  "timestamp": "2024-04-28T15:23:44.858+00:00",
  "status": 500,
  "error": "Internal Server Error",
  "message": "",
  "path": "/api/v1/students/858"
}
```

Výsledek:

Test v pořádku (očekávaný = reálný výsledek)

Nelze smazat již smazaného studenta a systém to rozpozná

3 EXEKUCE TESTŮ

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

4 BUG REPORT

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