

# FINÁLNÍ PROJEKT

## č.1



# ENGETO

Autor: Petra Zemanová  
Datum: 5.7.2024

## OBSAH

|   |    |
|---|----|
| ZADÁNÍ  | 3  |
| 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.. | 3  |
| Přístupové údaje:   | 3  |
| Poznámky:   | 3  |
| TEST SCENARIOS  | 4  |
| EXECUTION OF TESTS  | 8  |
| BUG REPORT  | 13 |

# 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 |  |
| REST-API |  |

## 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.

# TEST SCENARIOS

*Based on these test scenarios, I have verified the functionality of the application.*

## GET Request

### 1. Retrieve an existing student:

- **Precondition:**

Student with ID=609 exists in the database MySQLWorkbench.

```
SELECT * FROM student where id=609;
```

```
'609','17','j.luskgmail.cz','Jan','LUSK'
```

- **Steps:**

1. Create a GET request to the URL <http://108.143.193.45:8080/api/v1/students/609> in Postman.

2. Click the "Send" button.

- **Expected Result:**

Status code is 200 OK.

Response contains the following:

```
{  
  "id": 609,  
  "firstName": "Jan",  
  "lastName": "LUSK",  
  "email": "j.luskgmail.cz",  
  "age": 17  
}
```

### 2. Retrieve a non-existing student:

- **Precondition:**

Student with ID 1000 does not exist in the database MySQLWorkbench.

- **Steps:**

1. Create a GET request to the URL <http://108.143.193.45:8080/api/v1/students/1000> in Postman.

2. Click the "Send" button.

- **Expected Result:**

Status code is 404 Not Found.

## DELETE Request

### 1. Delete an existing student:

- **Precondition:**

Student with ID 506 exists in the database MySQLWorkbench..

- **Steps:**

1. Create a DELETE request to the URL <http://108.143.193.45:8080/api/v1/students/506> in Postman.
2. Click the "Send" button.

- **Expected Result:**

1. Status code is 200 OK.
2. Verify that the student with ID 214 was deleted from the database MySQLWorkbench using the SQL query `SELECT * FROM student where ID = 506.`

## **POST Request**

To successfully create a student:

- All fields are mandatory (firstName, lastName, email, age).
  - If a field is missing -> 400 Bad Request (lastName, email are mandatory).
- 1. All attributes are stored in the database in lowerCase() format:
  - firstName, lastName, email.
  - Example: StEfAn -> stored as Stefan.
  - Example: TeStEr123@GmAiL.CoM -> stored as tester123@gmail.com.
- 3. Field validation:
  - **age:** Must be between 1 and 150 years. If not -> 400 Bad Request age must be between 1-150.
  - **firstName, lastName:** Must be between 3 and 50 characters. If not -> 400 Bad Request firstName must be between 3-50 characters.
  - **email:** Must follow the format @.\*.
    - Examples:
      - stefan123@hotmail.com
      - 234234@test.hu
      - ieirerer@yourmail.cz
  - If not -> 400 Bad Request email must be in format @.\*xy.

### **1. Create a new student with correct data:**

- **Precondition:**

The database can accept new records.

- **Steps:**

1. Create a POST request to the URL <http://108.143.193.45:8080/api/v1/students> in Postman.
2. In the Body tab, select "raw" and "JSON" and insert the following JSON:

```
{
  "firstName": "MuMiNEk",
  "lastName": "Roztomily",
  "email": "muminek@roztomily.com",
  "age": 61
}
```
3. Click the "Send" button.

- **Expected Result:**

1. Status code is 201 Created.
2. Response contains the new student record with an assigned ID:

```
{
  "firstName": "Muminek",
  "lastName": "ROZTOMILY",
  "email": "muminek@roztomily.com",
  "age": 61
}
```
3. Verify that the new student was saved in the database MySQLWorkbench using the SQL query: `SELECT * FROM student WHERE id = <assigned_id>;`

## **2. Create a student with missing and invalid attributes:**

- **Precondition:**

The attributes lastName and email are mandatory, and age must be between 1 and 150.

- **Steps:**

1. Create a POST request to the URL <http://108.143.193.45:8080/api/v1/students> in Postman.
2. In the Body tab, select "raw" and "JSON" and insert the following JSON (missing lastName and invalid email and age):

```
{
  "firstName": "Mumínek",
  "email": "muminekroztomily.com",
  "age": -15
}
```
3. Click the "Send" button.

- **Expected Result:**

1. Status code is 400 Bad Request.

## **3. Create a user with invalid firstName and lastName lengths:**

- **Precondition:**

firstName and lastName must be between 3 and 50 characters.

- **Steps:**

1. Create a POST request to the URL <http://108.143.193.45:8080/api/v1/students> in Postman.
2. In the Body tab, select "raw" and "JSON" and insert the following JSON (too short firstName and too long lastName):

```
{
  "firstName": "Já",
  "lastName": "JsemRoztomilyMuminekCoSiNerikaROZTOMILYmuminekPriTeStování",
  "email": "muminek@neroztomily.com",
  "age": 60
}
```

}

3. Click the "Send" button.

- **Expected Result:**

1. Status code is 400 Bad Request.

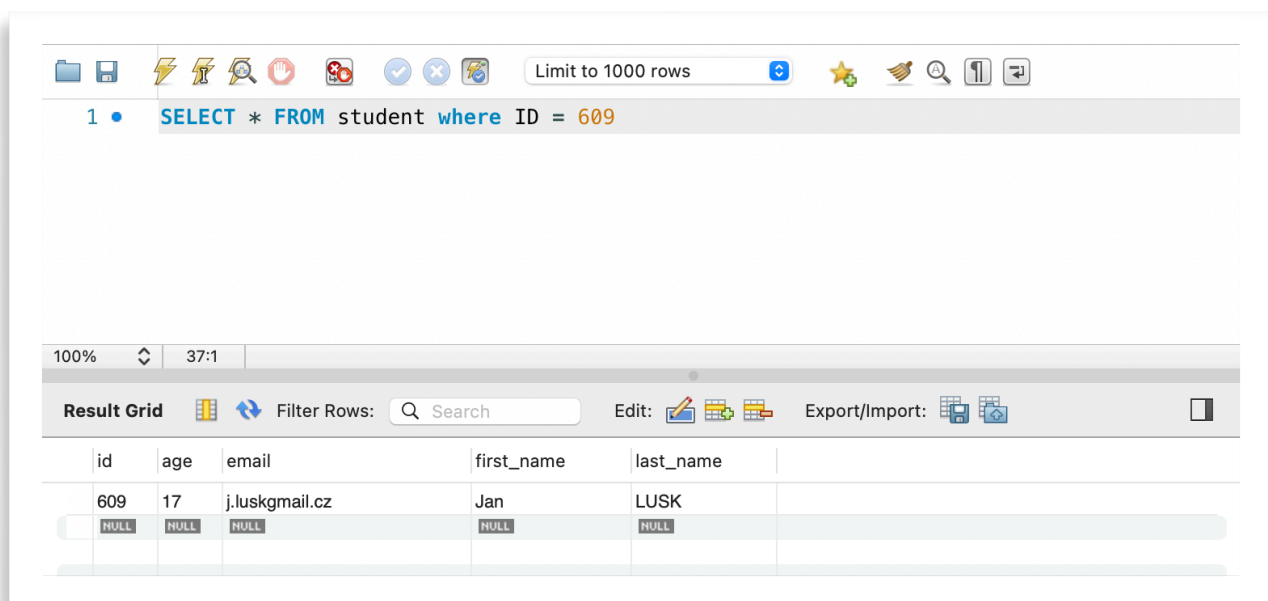
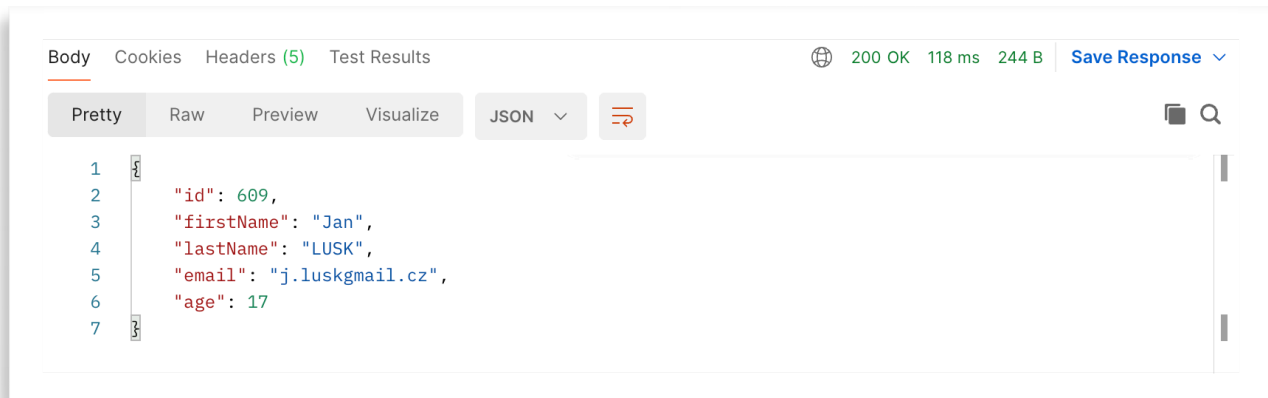
# EXECUTION OF TESTS

*I have run the test scenarios and attached the test results.*

## GET Request

### 1. Retrieve an existing student:

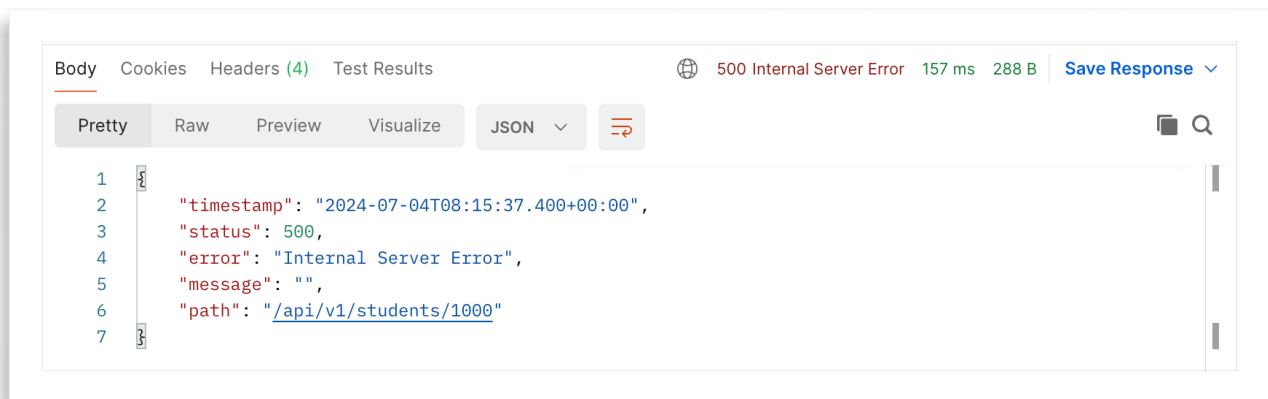
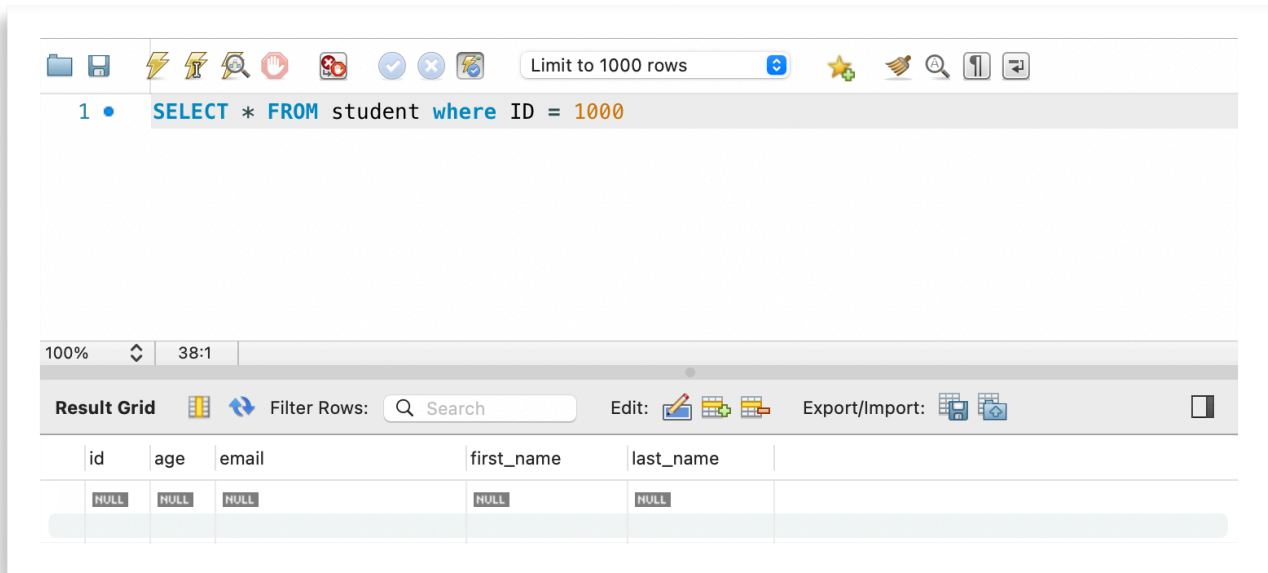
Student was found, status code **200 OK**



### 2. Retrieve a non-existing student:

Student was not found, the expected result was status code 404 not found, but an error status code **500 Internal Server Error** was displayed



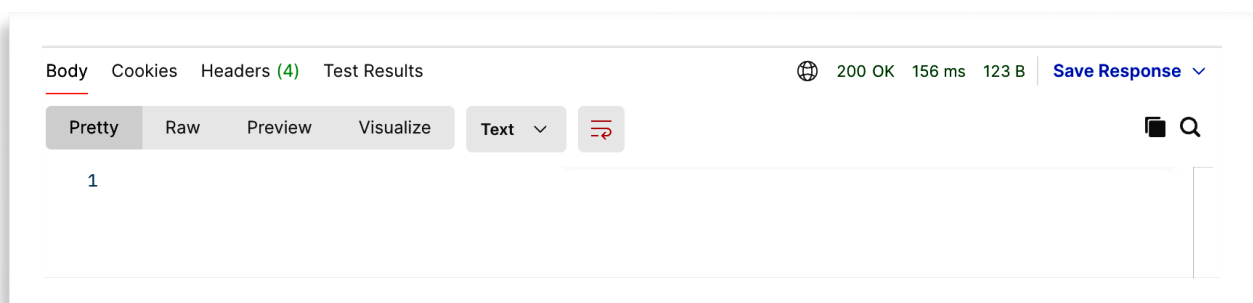


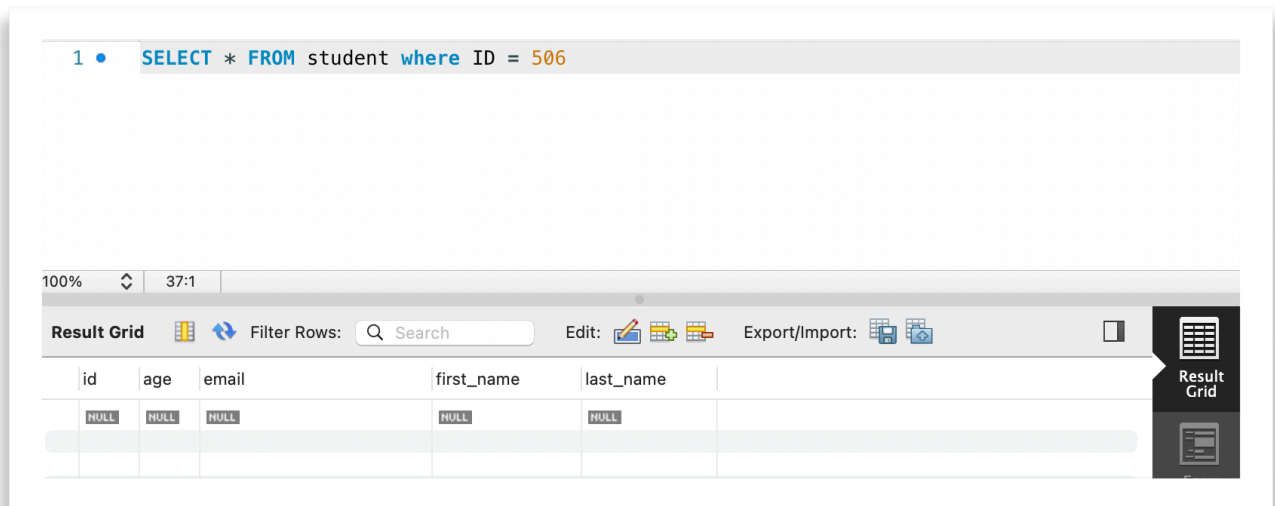
## DELETE Request

### 1. Delete an existing student:

Status code 200 OK.

Student was not found in database Postman.



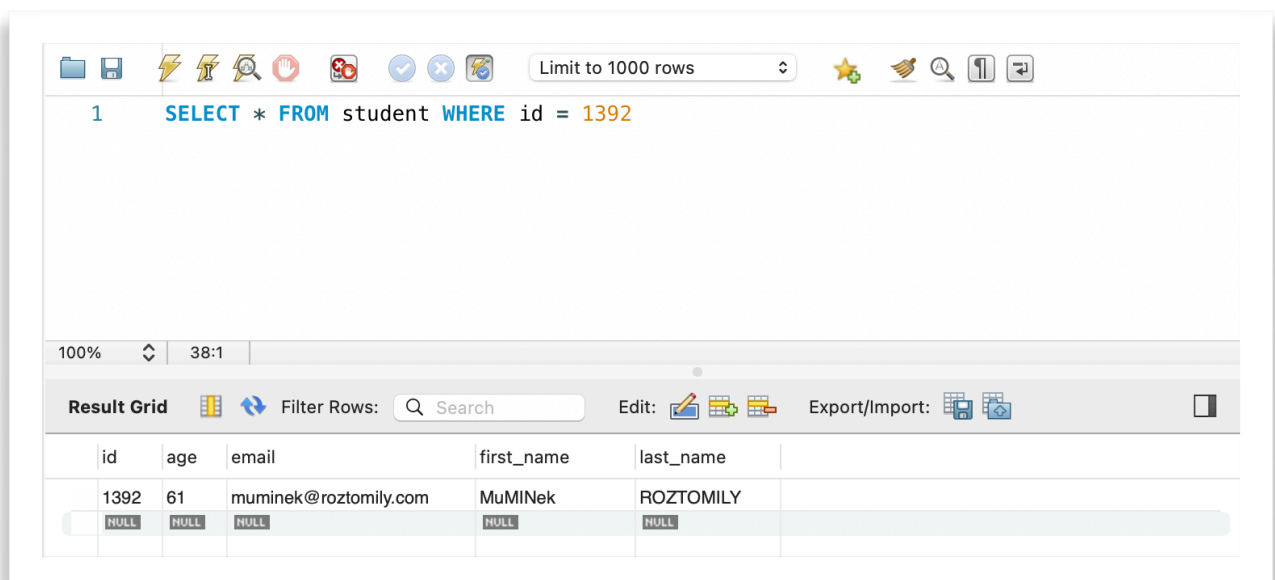


## POST Request

### 1. Create a new student with correct data:

Student with new ID 1392 was create, unfortunately format of firstName is not correct. Student was found in database MySQLWorkbench.

```
{
  "id": 1392,
  "firstName": "MuMINek",
  "lastName": "ROZTOMILY",
  "email": "muminek@roztomily.com",
  "age": 61
}
```



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 Beautify

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "firstName": "MuMiNeK",
3   "lastName": "Roztomily",
4   "email": "muminek@roztomily.com",
5   "age": 61
6 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 247 ms Size: 261 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1392,
3   "firstName": "MuMiNeK",
4   "lastName": "ROZTOMILY",
5   "email": "muminek@roztomily.com",
6   "age": 61
7 }
```

## 2. Create a student with missing and invalid attributes:

Student with new ID wasn't created, response in Postman contains error messages 400.

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 Beautify

none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "firstName": "Muminek",
3   "email": "muminekroztomily.com",
4   "age": -15
5 }
```

Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 59 ms Size: 263 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2024-07-05T17:57:40.971+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "message": "",
6   "path": "/api/v1/students"
7 }
```

### 3. Create a user with invalid firstName and lastName lengths:

Student with new ID wasn't created, response in Postman contains error messages 400.

The image shows a Postman interface for a REST client. The URL bar shows `http://108.143.193.45:8080/api/v1/students`. The method is `POST`. The body is set to `raw` and contains the following JSON:

```
{
  "firstName": "Já",
  "lastName": "JsemRoztomilyMuminekCoSiNerikaROZTOMILymuminekPriTeStováni",
  "email": "muminek@neroztomily.com",
  "age": 60
}
```

The response status is `400 Bad Request`, with a time of `55 ms` and a size of `263 B`. The response body is shown in the `Body` tab, formatted as JSON:

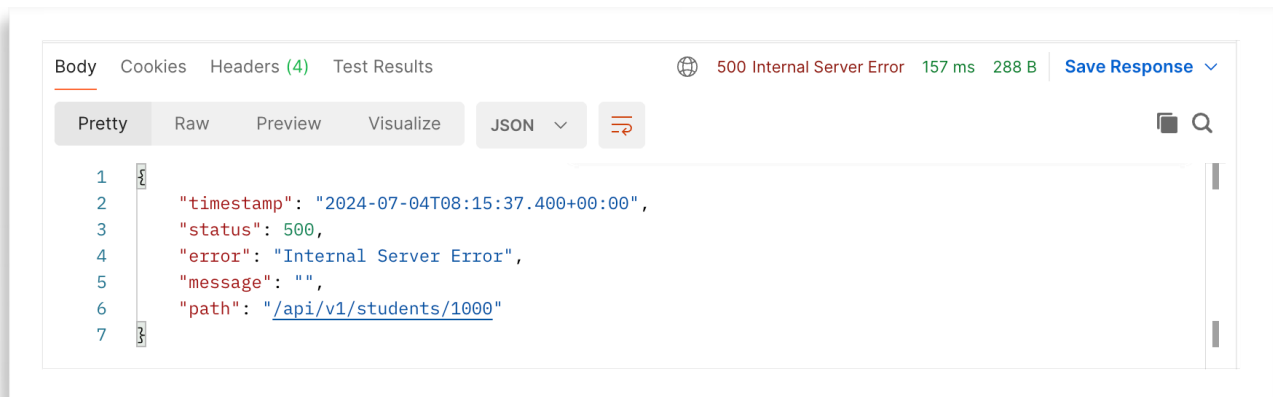
```
{
  "timestamp": "2024-07-05T18:12:49.460+00:00",
  "status": 400,
  "error": "Bad Request",
  "message": "",
  "path": "/api/v1/students"
}
```

# BUG REPORT

*Based on the scenarios I have run, I have discovered the following application errors.*

### GET Request -> 2. Retrieve a non-existing student:

Student was not found, the expected result was status code 404 not found, but an error status code **500 Internal Server Error** was displayed



**POST Request -> 1. Create a new student with correct data:**

All attributes should be stored in the database in lowerCase() format, unfortunately format is not correct.

