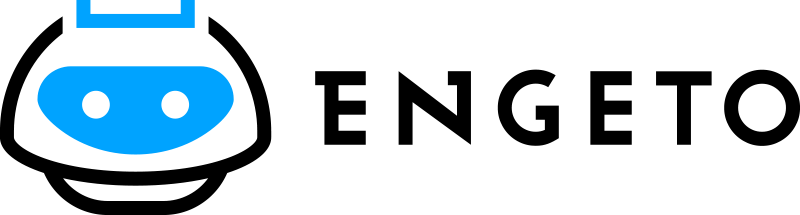
FINÁLNÍ PROJEKT   
 č.1



Autor: Michal Seibert  
Datum: 12. 6. 2024

OBSAH

[**ZADÁNÍ**](#_heading=h.30j0zll) **3**

[**TESTOVACÍ SCÉNÁŘE**](#_heading=h.2et92p0) **4**

[**EXEKUCE TESTŮ**](#_heading=h.tyjcwt) **5**

[**BUG REPORT**](#_heading=h.3dy6vkm) **5**

# 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: viz Discord Password: viz Discord |
| 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

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

There are certain requirements for database record attributes defined by our mentor in the video lesson:

* All attributes are mandatory (*age, email, first\_name, last\_name*)
* All attributes are stored in DB in lowercase format
* Attribute *age* is integer in range 1 - 150
* Attributes first\_name and *last\_name* are strings with length in range 3 - 50
* Attribute *email* is checked if it meets condition for valid email. There is a mask defined for checking: \*@\*.\*

## Scenario series 1 – Test GET request

### Abstract: Acquire data about the student who is identified by id of the record.

### For the purpose of testing this scenario, I tried to create a test student record first. Since I don't have sufficient rights in MySQL Workbench to execute the INSERT INTO statement, I tried to use the REST API to create the test student record. These are the JSON parameters of this student:

### {

### "id": 1190, "firstName": "josef",

### "lastName": "švejk",

### "email": "josef.svejk@rakousko.uhersko",

### "age": 35

### }

### I checked if the mentioned record really exists in database by SQL query: SELECT \* FROM student WHERE id = 1190 and result is shown below:

### Obsah obrázku text, software, Multimediální software, snímek obrazovky Popis byl vytvořen automaticky

Since this record does not meet the lowercase requirement defined at the beginning of this chapter, I tried to manually modify the last\_name column in MySQL Workbench to the value svejk. When I tried to verify the change, there was an uppercase value again because the database wouldn't let me do an UPDATE statement as well. As a result of this situation, I cannot work with a record that meets the defined criteria. I will take this fact into account later in the POST testing scenario. Now I will work with this "non-compliant" record:

## Scenario 1.a – Test GET request with existing id

|  |  |
| --- | --- |
| STEPS | Documentation |
| Verify in *MySQL Workbrench* that the record with id=1190 does exist in DB. | Obsah obrázku text, snímek obrazovky, software, Multimediální software  Popis byl vytvořen automaticky |
| In *Postman* select GET request, use address: *http://108.143.193.45:8080/api/v1/students/1190*  and press Send button. |  |

## Scenario 1.b – Test GET request with non-existing id

|  |  |
| --- | --- |
| STEPS | Documentation |
| Verify in *MySQL Workbrench* that the record with id=214 does exist in DB. | Obsah obrázku snímek obrazovky, text, software, Multimediální software  Popis byl vytvořen automatickyObsah obrázku text, snímek obrazovky, Písmo  Popis byl vytvořen automaticky |
| In *Postman* select GET request, use address: *http://108.143.193.45:8080/api/v1/students/1190*  and press Send button. |  |

## Scenario 1.c – Test GET request with string input

|  |  |
| --- | --- |
| STEPS | Documentation |
| In *Postman* select GET request, use address: *http://108.143.193.45:8080/api/v1/students/@b12c$1* and press Send button.  Reason: check if there is not some kind of parser which separate numbers and try too check “relevant” record. |  |

## Scenario 1.d – Test GET request with empty input

|  |  |
| --- | --- |
| STEPS | Documentation |
| In *Postman* select GET request, use address: *http://108.143.193.45:8080/api/v1/students/* and press Send button.  Reason: how the application react on empty input? |  |

## Scenario series 2 – Test POST request

### Abstract: Create new student in database.

## Scenario 2.a – Test POST request with correct input data

|  |  |
| --- | --- |
| STEPS | Documentation |
| In *Postman* select POST request, use address: *http://108.143.193.45:8080/api/v1/students/1190* |  |
| In *Postman*, set the body of the message in JSON format with 4 required attributes whose values ​​meet the requirements stated at the beginning of this chapter.  Press Send button. | Obsah obrázku text, software, Multimediální software, Grafický software  Popis byl vytvořen automaticky |

## Scenario 2.b – Test POST request with correct input data, but same as existing user

|  |  |
| --- | --- |
| STEPS | Documentation |
| In *Postman* select POST request, use address: *http://108.143.193.45:8080/api/v1/students/1190* |  |
| In *Postman*, set the body of the message in JSON format with 4 required attributes whose values ​​meet the requirements stated at the beginning of this chapter.  Press Send button. | Obsah obrázku text, software, Multimediální software, snímek obrazovky  Popis byl vytvořen automaticky |

## Scenario 2.c – Test POST request with empty data

|  |  |
| --- | --- |
| STEPS | Documentation |
| In *Postman* select POST request, use address: *http://108.143.193.45:8080/api/v1/students/1190* |  |
| In *Postman*, set the empty body of the message.  Press Send button. | Obsah obrázku snímek obrazovky, text, software, Multimediální software  Popis byl vytvořen automaticky |

## Scenario 2.d – Test POST request with missing one of attribute in data collection input

|  |  |
| --- | --- |
| STEPS | Documentation |
| In *Postman* select POST request, use address: *http://108.143.193.45:8080/api/v1/students/1190* |  |
| In *Postman*, set the body of the message in JSON format so that exactly one of the required attributes is always missing in four consecutive tests. {"lastName": "švejk","email": "josef.svejk@rakousko.uhersko","age": 35}{  "firstName": "josef","email": "josef.svejk@rakousko.uhersko","age": 35}{  "firstName": "josef","lastName": "švejk","age": 35}{  "firstName": "josef","lastName": "švejk","email": "josef.svejk@rakousko.uhersko} Press Send button for each of the separate tests. | Obsah obrázku text, software, Multimediální software, Grafický software  Popis byl vytvořen automaticky |

## Scenario 2.e – Test POST request with missing two of attributes in data collection input

|  |  |
| --- | --- |
| STEPS | Documentation |
| In *Postman* select POST request, use address: *http://108.143.193.45:8080/api/v1/students/1190* |  |
| In *Postman*, set the body of the message in JSON format so that exactly one of the required attributes is always missing in five consecutive tests. {"email": "josef.svejk@rakousko.uhersko","age": 35}{  "firstName": "josef","age": 35}{  "firstName": "josef","lastName": "švejk",}{  "firstName": "josef","email": "josef.svejk@rakousko.uhersko}{  "lastName": "švejk","age": 35} Press Send button for each of the separate tests. | Obsah obrázku snímek obrazovky, text, software, Multimediální software  Popis byl vytvořen automaticky |

## Scenario 2.f – Test POST request with only one attribute in data collection input

|  |  |
| --- | --- |
| STEPS | Documentation |
| In *Postman* select POST request, use address: *http://108.143.193.45:8080/api/v1/students/1190* |  |
| In *Postman*, set the body of the message in JSON format so that exactly one of the required attributes is always missing in four consecutive tests. {"firstName": "josef"}{  "lastName": "švejk"}{"email": "josef.svejk@rakousko.uhersko}{"age": 35} Press Send button for each of the separate tests. | Obsah obrázku snímek obrazovky, text, software, Multimediální software  Popis byl vytvořen automaticky |

## Scenario 2.g – A test POST request with all attributes not meeting their defined criteria

|  |  |
| --- | --- |
| STEPS | Documentation |
| In *Postman* select POST request, use address: *http://108.143.193.45:8080/api/v1/students/1190* |  |
| In *Postman*, set the body of the message in JSON format so that only first\_name attribute is failing the criteria and rest tree attributes are meets their criteria. {"firstName": "","lastName": "","email": "","age": 0 } {"firstName": "abcdefghijabcdefghijabcdefghijabcdefghijabcdefghija","lastName": "abcdefghijabcdefghijabcdefghijabcdefghijabcdefghija","email": "abc","age": "151" } Press Send button for each of the separate tests. | Obsah obrázku snímek obrazovky, text, software, Multimediální software  Popis byl vytvořen automaticky |

# EXEKUCE TESTŮ

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

## Scenario 1.a – Test GET request with existing id

|  |  |
| --- | --- |
| Expectation | Result |
| Status: 200 OK  Respective list of the aforementioned attributes and its values in response:  {"id": 1190,  "firstName": "josef","lastName": "ŠVEJK","email": "josef.svejk@rakousko.uhersko","age": 35} | Obsah obrázku text, snímek obrazovky, software, Multimediální software  Popis byl vytvořen automaticky |
| Validation | PASS |
|  | |

## Scenario 1.b – Test GET request with non-existing id

|  |  |
| --- | --- |
| Expectation | Result |
| Status: 404 Not Found  No data at all in response. | Obsah obrázku text, snímek obrazovky, software, Multimediální software  Popis byl vytvořen automaticky |
| Validation | FAIL |
| Status is 500 Internal Server Error, not 404.  Note: There are 5 attributes in response, non-regarding the database, but to the state of the response. | |

## Scenario 1.c – Test GET request with string input

|  |  |
| --- | --- |
| Expectation | Result |
| Status: 400 Bad Request  No data at all in response. | Obsah obrázku text, snímek obrazovky, software, Multimediální software  Popis byl vytvořen automaticky |
| Validation | PASS |
| Note: There are 5 attributes in response, non-regarding the database, but to the state of the response. | |

## Scenario 1.d – Test GET request with empty input

|  |  |
| --- | --- |
| Expectation | Result |
| Status: 400 Bad Request  No data at all in response. versus  Status: 200 OK  Return in response data about all the students. | Obsah obrázku text, snímek obrazovky, software, Multimediální software  Popis byl vytvořen automaticky |
| Validation | UNCLEAR |
| Is it the correct behavior (to return all students), or is there a data leak? | |

## Scenario 2.a – Test POST request with correct input data

|  |  |
| --- | --- |
| Expectation | Result |
| Status: 200 OK  Data about created student in response. | Obsah obrázku text, snímek obrazovky, software, Multimediální software  Popis byl vytvořen automaticky |
| Validation | PASS |
|  | |

## Scenario 2.b – Test POST request with correct input data, but same as existing user

|  |  |
| --- | --- |
| Expectation | Result |
| Status: 200 OK  Data about created student in response. | Obsah obrázku text, snímek obrazovky, software, Multimediální software  Popis byl vytvořen automaticky |
| Validation | PASS |
|  | |

# BUG REPORT

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

## Bugs

|  |  |
| --- | --- |
| Scenario | BUG |
| 1.b – Test GET request with non-existing id | Status 500 returned, not 404. |
| 1.d – Test GET request with empty input | Data about all students returned. |

## Unclear

|  |  |
| --- | --- |
| Scenario | Situation |
| 1.d – Test GET request with empty input | Data about all students returned. |