FINÁLNÍ PROJEKT   
  č.1



Autor: Ing.Nikola Knap

Datum: 1.6.2024

**OBSAH**

[**ZADÁNÍ 3**](https://docs.google.com/document/d/11v5l9NI4n8G6Xc67Q-po_R4vw0SM_XezYu1Wpd9eVUg/edit#heading=h.30j0zll)

[**TESTOVACÍ SCÉNÁŘE 4**](https://docs.google.com/document/d/11v5l9NI4n8G6Xc67Q-po_R4vw0SM_XezYu1Wpd9eVUg/edit#heading=h.2et92p0)

[**EXEKUCE  TESTŮ 5**](https://docs.google.com/document/d/11v5l9NI4n8G6Xc67Q-po_R4vw0SM_XezYu1Wpd9eVUg/edit#heading=h.tyjcwt)

**BUG REPORT 13**

**ZADÁNÍ**

Cílem prvního projektu je otestování REST rozhraní. Rozhraní slouží pro manipulaci dat studentů. Otestujte metody GET, POST a DELETE. Nezapomeňte aplikovat znalosti z lekcí 1 až 4 ( testovací scénáře, defekt, status kódy, SQL dotazy,...)

Platí, že:

* metoda GET zobrazí data o existujícím studentovi
* metoda POST založí záznam o studentovi
* metoda DELET smaže data o studentovi

Potrebne aplikace: Postman, MySQLWorkbench, MS Word

Přístupové údaje:

|  |  |
| --- | --- |
| Databáze | Database: qa\_demo  Host: aws.connect.psdb.cloud  userName: q524i16o56a3hjktx53r  Password: pscale\_pw\_JvkHoSxuIxh8GLHYEDCzCvgOqyAgkU7zfNspWwhMoCX |
| REST-API | http://108.143.193.45:8080/api/v1/students/ |

**TESTOVACÍ SCÉNÁŘE**

**Metoda GET:**

* vyhledani studenta s nahodnym id
* vyhledani studenta s neexistujicim id
* vyhledani studenta s preklepem v prikazu
* vyhledani studenta podle veku

**Metoda DELETE:**

* smazani studenta
* smazani jiz smazaneho studenta

**Metoda GET:**

* zalozeni noveho studenta
* zalozeni noveho studenta s chybnym formatem emailu

**EXEKUCE  TESTŮ**

1. **METODA – GET**
   1. TEST

**Testovaci scenar:** Vyhledani studenta s nahodnym id, napr. id=242

|  |  |
| --- | --- |
| MySQLWorkbench | Postman |
|  |  |

**Skutecny výsledek:**

MySQLWorkbench – zobrazi se informace o studentovi pod timto id (age, email, first\_name, last\_name)

Postman – zobrazi se informace o studentovi pod timto id

**Ocekavany výsledek:**

MySQLWorkbench – odpovida skutecnemu vysledku

Postman – odpovida skutecnemu vysledku

* 1. TEST

**Testovaci scenar:** Vyhledani studenta s id=244 (neexistujici student)

|  |  |
| --- | --- |
| MySQLWorkbench | Postman |
|  |  |

**Skutecny výsledek:**

MySQLWorkbench - nebyl nalezen zadny vystup

Postman - vystup zaznamenal jako status kod: 500 Internal Server Error

**Ocekavany výsledek:**

MySQLWorkbench – odpovida skutecnemu vysledku

Postman – status kod 404 Not Found

**Poznamka:** 1.2 TEST bude evidovan jako defect

* 1. TEST

**Testovaci scenar:** Vyhledani studenta s id=2r2 (preklep v prikazu)

|  |  |
| --- | --- |
| MySQLWorkbench | Postman |
|  |  |

**Skutecny výsledek:**

MySQLWorkbench - dojde k upozorneni ve forme chybove hlasky

Postman - vystup zaznamenal jako status 400 Bad Reguest

**Ocekavany výsledek:**

MySQLWorkbench – odpovida skutecnemu vysledku

Postman – odpovida skutecnemu vysledku

* 1. TEST

**Testovaci scenar:** Vyhledani studenta s vekem 34 (age=34)

|  |  |
| --- | --- |
| MySQLWorkbench | Postman |
|  |  |

**Skutecny výsledek:**

MySQLWorkbench - dojde k vyhledani všech studentu, kteří mají ve sloupci age číslo 34. Response se vygeneruje 45 row (celkovy pocet nalezenych vysledku)

Postman - vystup zaznamenal jako status 404 Not Found

**Ocekavany výsledek:**

MySQLWorkbench – odpovida skutecnemu vysledku

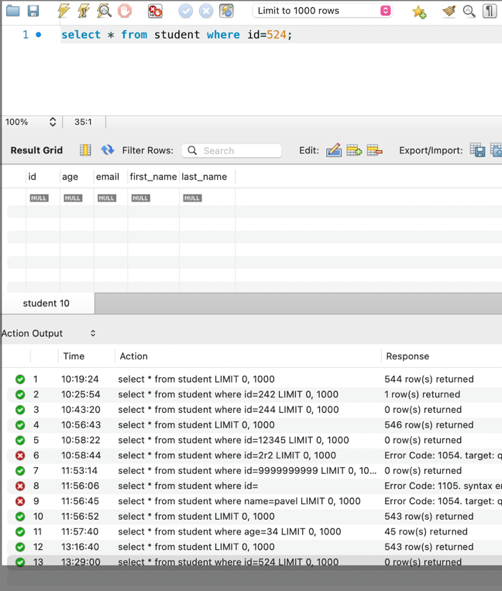
Postman – aplikace nic nevyhleda

**Poznamka:** 1.4 TEST bude evidovan jako defect

1. **METODA – DELETE**
   1. TEST

**Testovaci scenar:** Priklad smazani studenta id=524

|  |  |
| --- | --- |
| MySQLWorkbench | Postman |
|  |  |

**Skutecny výsledek:** 

**Ocekavany výsledek:**

MySQLWorkbench – odpovida skutecnemu vysledku

Postman – odpovida skutecnemu vysledku

* 1. TEST

**Testovaci scenar:** Priklad smazani jiz smazaného studenta id=524

|  |  |
| --- | --- |
| MySQLWorkbench | Postman |
|  |  |

**Skutecny výsledek:**

MySQLWorkbench – dojde k vyhledani 0 studentu (student s timto id se v databazi nenachazi)

Postman – vystup zaznamenal jako status 500 Inernal Server Error

**Ocekavany výsledek:**

MySQLWorkbench - odpovida skutečnému vysledku

Postman – status kod by mel byt 404 Not Found

**Poznamka:** 2.2 TEST bude evidovan jako defect

1. **METODA – POST** 
   1. TEST

**Testovaci scenar:** Zalozeni noveho studenta

|  |  |
| --- | --- |
| Postman | MySQLWorkbench |
|  |  |

**Skutecny výsledek:**

Postman – pri zalozeni noveho studenta se telo kodu doplni o nahodne id=1121 a v casti lastName dojde ke zmene velikosti pisma

MySQLWorkbench – pri vyhledani nove zalozeneho studenta pri vygenerovanem id=1121 aplikace najde prislusne informace presne aplikace vygenerovala – cili se zmenou velikosti pisma v last\_name

**Ocekavany výsledek:**

Postman – aplikace studenta zalozi a prideli mu id, presne dle zadani ponese informace o studentovi jako LowerCase (vse malym pismem)

MySQLWorkbench – aplikace data nacte presne jak byla zadana pri zalozeni

**Poznamka:** 3.1 TEST bude evidovan jako defect

* 1. TEST

**Testovaci scenar:** Zalozeni noveho studenta s chybnym formatem emailu

|  |  |
| --- | --- |
| Postman | MySQLWorkbench |
|  |  |

**Skutecny výsledek:**

Postman – pri zalozeni noveho studenta se telo kodu doplni o nahodne id=1123, aplikace nereaguje na spatne zadany format emailu

MySQLWorkbench – pri vyhledani nove zalozeneho studenta pri vygenerovanem id=1123 aplikace najde prislusne informace presne, ktere Postman aplikace vygenerovala – nebere v potaz spatne zadany format emailu

**Ocekavany výsledek:**

Postman – aplikace upozorni na spatne zadany format emailu

MySQLWorkbench – aplikace data nacte presne jak byla zadana pri zalozeni

**Poznamka:** 3.2 TEST bude evidovan jako defect

BUG REPORT

*Dle meho otestovani database studentu jsem dopela k temto defectum:*

1.2 TEST - vyhledani studenta s neexistujicim id

1.4 TEST - vyhledani studenta podle veku

2.2 TEST - smazani jiz smazaneho studenta

3.1 TEST - zalozeni noveho studenta

3.2 TEST - zalozeni noveho studenta s chybnym formatem emailu