FINÁLNÍ PROJEKT   
 č.1



Autor: Zdeněk Dvořáček

Datum: 6.10.2024

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: \*\*\*\*\*  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 SCÉNÁŘ GET č.1

1. Testovací scénář GET

Abstract: Get data about existing student

Test data preparation – existing student id=367

select \* FROM student where id=367;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

# id, age, email, first\_name, last\_name

'367', '80', 'alfred@mail.com', 'alfred', 'HIW'

Steps:

1. GET <http://108.143.193.45:8080/api/v1/students/367>
2. Press send button

Expected Result:

1. Status CODE 200 OK
2. Response

{

    "id": 367,

    "firstName": "alfred",

    "lastName": "HIW",

    "email": "alfred@mail.com",

    "age": 80

}

# TESTOVACÍ SCÉNÁŘ GET č.2

1. Testovací scénář GET

Abstract: Get data about non-existing student

Test data preparation – non-existing student id=214

select \* FROM student where id=214;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Steps:

1. GET <http://108.143.193.45:8080/api/v1/students/214>
2. Press send button

Expected Result:

1. Status CODE 404

# TESTOVACÍ SCÉNÁŘ DELETE č.3

1. Testovací scénář DELETE
2. Abstract: DELETE data about existing student

Test data preparation – existing student id=367

select \* FROM student where id=367;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

# id, age, email, first\_name, last\_name

'367', '80', 'alfred@mail.com', 'alfred', 'HIW'

Steps:

1. DELETE <http://108.143.193.45:8080/api/v1/students/367>
2. Press send button

Expected Result:

1. Status CODE 200 OK

# TESTOVACÍ SCÉNÁŘ DELETE č.4

1. Testovací scénář DELETE
2. Abstract: DELETE data about non-existing student

Test data preparation – non-existing student id=367

select \* FROM student where id=367;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

id, age, email, first\_name, last\_name

Steps:

1. DELETE <http://108.143.193.45:8080/api/v1/students/367>
2. Press send button

Expected Result:

1. Status CODE 404

# TESTOVACÍ SCÉNÁŘ POST č.5

1. Testovací scénář POST
2. Abstract: POST Úspěšné založení studenta – vyhovující validačním kritériím

Steps:

1. Request body JSON:

{

    "firstName": "Igor",

    "lastName": "Buřt",

    "email": "burtik1@mail.com",

    "age": 120

}

1. Press send button

Expected Results:

1. Status code: 201 created
2. Response:

{

    "id": 1853,

    "firstName": "Igor",

    "lastName": "BUŘT",

   "email": "burtik1@mail.com",

    "age": 120

}

1. Database check: Kontrola dat podle přiřazeného id

Validační kritéria:

1. Všechny atributy jsou povinné (firstName, lastName, email, age). Pokud atribut chybí, 400 Bad request (+ info jaké povinné atributy chybí)
2. Všechny atributy se ukládají do databáze lowerCase().
3. Kritéria hodnot atributů:
   1. Age v rozmezí 1-150 (jinak 400 Bad request – Age must be between 1-150)
   2. firstName, lastName v rozmezí 3-50 znaků (jinak 400 Bad request – firstName,lastName must be between 3-50 characters)
   3. Email musí obsahovat „@“, „.“ Jinak 400 Bad request: Email must be in format [\*@\*.\*](mailto:*@*.*)

# TESTOVACÍ SCÉNÁŘ POST č.6

1. Testovací scénář POST
2. Abstract: POST Úspěšné nezaložení studenta – nevyhovující validačním kritériím

Steps:

1. Request body JSON:

{

    "firstName": "Igor",

    "lastName": "Bu",

    "email": "burtik1mail.com",

    "age": 151

}

1. Press send button

Expected Results:

1. Status code:

400 Bad request – „Age must be between 1-150,

400 Bad request – lastName must be between 3-50 characters,

400 Bad request – Email must be in forma \*@\*.\*

1. Response:

{

   "id": 1854,

    "firstName": "Igor",

    "lastName": "BU",

    "email": "burtik1mail.com",

    "age": 151

}

1. Database check: Kontrola dat podle přiřazeného id

Validační kritéria:

1. Všechny atributy jsou povinné (firstName, lastName, email, age). Pokud atribut chybí, 400 Bad request (+ info jaké povinné atributy chybí)
2. Všechny atributy se ukládají do databáze lowerCase().
3. Kritéria hodnot atributů:
   1. Age v rozmezí 1-150 (jinak 400 Bad request – Age must be between 1-150)
   2. firstName, lastName v rozmezí 3-50 znaků (jinak 400 Bad request – firstName,lastName must be between 3-50 characters)
   3. Email musí obsahovat „@“, „.“ Jinak 400 Bad request: Email must be in format [\*@\*.\*](mailto:*@*.*)

# EXEKUCE TESTŮ

1. Výsledek testovacího scénáře GET – Test prošel bez chyb

Obsah obrázku snímek obrazovky

Popis byl vytvořen automaticky

1. Výsledek testovacího scénáře GET- Neprošel – Chyba statusu 500 internal server error

Obsah obrázku text, snímek obrazovky

Popis byl vytvořen automaticky

1. Výsledek testovacího scénáře DELETE – TEST prošel status 200 OK

Obsah obrázku snímek obrazovky, text

Popis byl vytvořen automaticky

1. Výsledek testovacího scénáře DELETE – TEST neprošel - Chyba statusu 500 internal server error

Obsah obrázku text, software, řada/pruh, číslo

Popis byl vytvořen automaticky

1. Výsledek testovacího scénáře POST – Test neprošel – Chyba statusu 200 ok

Chyby v databázi:

firstName, lastName není lowerCase()

Obsah obrázku text, software, řada/pruh, snímek obrazovky

Popis byl vytvořen automaticky

Obsah obrázku text, snímek obrazovky, software, Počítačová ikona

Popis byl vytvořen automaticky

1. Výsledek testovacího scénáře POST – Test neprošel – Chyba statusu 200 ok

Obsah obrázku text, řada/pruh, snímek obrazovky

Popis byl vytvořen automaticky

I přes validační kritéria se záznam uložil do databáze s chybami:

firstName, lastName není lowerCase()

nenastala chyba 400 Bad request pro age

nenastala chyba 400 Bad request pro email

nenastala chyba 400 Bad request pro počet znaků v lastName

Obsah obrázku text, snímek obrazovky, software, Počítačová ikona

Popis byl vytvořen automaticky

# BUG REPORT

Na základě **2.** výsledku testovacího scénáře GET provedeného scénáře jsem dostal chybovou hlášku status 500 internal server error místo očekávaného Status CODE 404.

Na základě **4.** výsledku testovacího scénáře DELETE provedeného scénáře jsem dostal chybovou hlášku status 500 internal server error místo očekávaného Status CODE 404.

Na základě **5.** výsledku testovacího scénáře POST provedeného scénáře jsem dostal STATUS 200 ok, ale měl být 201 ok. Dále chyby v databázi firstName, lastName není loweCase.

Na základě **6**. výsledku testovacího scénáře POST provedeného scénáře jsem dostal STATUS 200 ok, ale měl být 201 ok. Dále chyby:

I přes validační kritéria se záznam uložil do databáze s chybami:

1. firstName, lastName není lowerCase()
2. nenastala chyba 400 Bad request pro age
3. nenastala chyba 400 Bad request pro email
4. nenastala chyba 400 Bad request pro počet znaků v lastName