FINÁLNÍ PROJEKT   
   č.1



Autor: Miroslava Krblichová

Uživatelské jméno Discord: Miru\*

Datum: 8. 8. 2024

OBSAH

**[ZADÁNÍ](https://docs.google.com/document/d/1iRXztIY53d848KiMKNXpGAJ4yt72p3VJNk1j57S5ZwQ/edit" \l "heading=h.30j0zll)**

**[TESTOVACÍ SCÉNÁŘE](https://docs.google.com/document/d/1iRXztIY53d848KiMKNXpGAJ4yt72p3VJNk1j57S5ZwQ/edit" \l "heading=h.30j0zll)**

**[EXEKUCE  TESTŮ](https://docs.google.com/document/d/1iRXztIY53d848KiMKNXpGAJ4yt72p3VJNk1j57S5ZwQ/edit" \l "heading=h.30j0zll)**

**BUG REPORT**

**3**

**4**

**5 – 14**

**15 - 20**

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 SCÉNÁŘE**

***Na základě uvedených testovacích scénářů jsem ověřila funkčnost aplikace.***

ABSTACT:

- Metoda GET vypíše existujícího studenta s hraničním ID 319 nebo 1773.

- Metoda GET nevypíše neexistujícího studenta, ale vypíše špatný status.

- Metoda GET vypíše záznamy všech studentů ve správném formátu JSON.

- Metoda DELETE smaže existujícího studenta s hraničními hodnotami ID.

- Metoda DELETE správně vypíše chybu při mazání neexistujícího studenta, ale zvolí špatný status.

- Metoda DELETE vypíše chybu při mazání již smazaného studenta, ale se špatným statusem.

- Metoda POST vytvoří nového studenta v tabulce, ale se špatným statusem.

- Metoda POST vytvoří studenta s již existujícím ID a emailem a původního studenta přepíše.

- Metoda POST nevytvoří studenta s příliš dlouhým ID nebo s ID složeného ze speciálních či numerických znaků nebo písmenné ID nebo prázdnou hodnotu.

- Metoda POST vytvoří studenta i přesto, že jako firstName zadáme číselné hodnoty nebo hodnoty složené ze speciálních znaků nebo numerická znaménka nebo diakritická znaménka nebo nesmyslné jméno nebo velmi dlouhé hodnoty.

- Metoda POST vytvoří studenta i přesto, že jako firstName zadáme prázdnou hodnotu nebo mezeru.

- Metoda POST vytvoří studenta i přesto, že jako lastName zadáme prázdnou hodnotu nebo mezeru.

- Metoda POST vytvoří studenta i přesto, že jako lastName zadáme číselné hodnoty nebo hodnoty složené ze speciálních znaků nebo numerická znaménka nebo diakritická znaménka nebo nesmyslné jméno nebo prázdnou hodnotu nebo velmi dlouhé hodnoty.

- Metoda POST nevytvoří studenta, pokud jako age zadáme hodnoty složené ze speciálních znaků nebo numerická znaménka nebo diakritická znaménka nebo nesmyslný údaj nebo prázdnou hodnotu.

- Metoda POST vytvoří studenta při zadání emailu ve špatném formátu.

- Metoda POST vytvoří studenta při zadání emailu s mezerou nebo prázdnou hodnotou.

- Metoda POST vytvoří studenta při zadání již existujícího emailu.

- Metoda POST vytvoří studenta s nevyplněnými údaji u firstName a lastName.

- Metoda POST vytvoří studenta pod jiným ID, přestože v tabulce již existuje student se stejnými ostatními údaji.

- Metoda POST vytvoří studenta i s nesmyslným věkem (-7, 0, 1, 160) nebo s příliš dlouhou hodnotou.

- Metoda POST správně zaokrouhlí desetinné číslo na číslo celé u sloupce s ID a age.

- Postman správně vytvoří studenta přes metodu POST a vytáhne data z tabulky na Workbench za pomocí metody GET a studenta zase smaže pomocí metody DELETE.

- Metoda PUT neaktualizuje údaje z důvodu nepovoleného přístupu k této metodě, ale vypíše nevhodný status.

- Toto Rest API nepodporuje stránkování ani filtrování.

- Metoda POST vypíše výsledek v přijatelném čase.

- Metoda GET vypíše výsledek v přijatelném čase.

- Metoda DELETE smaže studenta v přijatelném čase.

**EXEKUCE  TESTŮ**

***Testovací scénáře jsem provedla, přikládám výsledky testů.***

*Tyto kroky se budou opakovat u každého testovacího případu:*

*Steps to reproduce:*

1. *Otevři MySQL workbench*
2. *V záložce query vlož požadavek SELECT \* FROM students a spusť*
3. *Otevři Postman*

*ABSTRACT:*

*Metoda GET vypíše existujícího studenta s hraničním ID 319 nebo 1773.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu GET*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání.*
3. *Na konec url adresy zadám studenta existujícího studenta s Id 319 nebo 1773*
4. *Vpravo nahoře zmáčknu SEND*
5. *Porovnám výsledky s tabulkou na workbench, jestli jsou stejné*

*Skutečný výsledek (Actual result):*

*Status 200: OK, vypsán student s ID 319 nebo 1773 a údaje o studentovi se shodují s tabulkou.*

*Očekáváný výsledek (Expected result):*

*Status 200: OK, vypsán student s ID 319 nebo 1773 a údaje o studentovi se shodují s tabulkou.*

*ABSTRACT:*

*Metoda GET nevypíše neexistujícího studenta, ale vypíše špatný status.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu GET*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání.*
3. *Na konec url adresy zadám neexistujícího studenta s Id 1*
4. *Vpravo nahoře zmáčknu SEND*
5. *Porovnám výsledky s tabulkou na workbench.*

*Skutečný výsledek (Actual result):*

***X****Status 500: Internal error, student nebyl nalezen.*

*Očekáváný výsledek (Expected result):*

*Status 404: Not found, nenalezeno.*

*ABSTRACT:*

*Metoda GET vypíše záznamy všech studentů ve správném formátu JSON.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu GET*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání.*
3. *Na konec URL adresy nezadám žádné ID studenta.*
4. *Vpravo nahoře zmáčknu SEND*
5. *Ve výsledku zkontroluji, jestli se vypsali všichni studenti od prvního do posledního údaje v tabulce ve WorkBench*
6. *Poté u výpisu výsledků vyberu hlavičku s názvem ‘Headers‘ a v tabulce, kde je Key ‘Content type‘, zkontroluji formát výsledků u Value.*

*Skutečný výsledek (Actual result):*

*Status OK: záznamy byly vypsány a to ve správném formátu JSON.*

*Očekáváný výsledek (Expected result):*

*Status OK: záznamy byly vypsány a to ve správném formátu JSON.*

*ABSTRACT:*

*Metoda DELETE smaže existujícího studenta s hraničními hodnotami 319 nebo 1773.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu DELETE*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání.*
3. *Za lomítko, na konec url adresy, zadám existujícího studenta s Id 319 nebo 1773*
4. *Porovnám výsledky s tabulkou na WorkBench, zda jestli byl student odstraněn.*

*Skutečný výsledek (Actual result):*

*Status 200: OK, Student byl vymazán.*

*Očekáváný výsledek (Expected result):*

*Status 200: OK, Student byl vymazán.*

*ABSTRACT:*

*Metoda DELETE správně vypíše chybu při mazání neexistujícího studenta, ale zvolí špatný status.*

*Steps to reproduce:*

1. *Na začátku příkazového řádku vyberu metodu DELETE*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání.*
3. *Za lomítko, na konec URL adresy, zadám (dle tabulky) neexistujícího studenta s Id 317*
4. *Porovnám výsledky s tabulkou na WorkBench, zda nedošlo ke změnám.*

*Skutečný výsledek (Actual result):*

***X****Status 500: Internal server error, student nebyl nalezen.*

*Očekáváný výsledek (Expected result):*

*Status 404: Not found, nenalezeno.*

*ABSTRACT:*

*Metoda DELETE vypíše chybu při mazání již smazaného studenta, ale se špatným statusem.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu DELETE*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání.*
3. *Za lomítko, na konec url adresy, zadám (dle tabulky) neexistujícího studenta s Id 1777*
4. *Porovnám výsledky s tabulkou na WorkBench, zda nedošlo ke změnám.*

*Skutečný výsledek (Actual result):*

***X****Status 500:Internal server error, student nebyl nalezen.*

*Očekáváný výsledek (Expected result):*

*Status 410: Gone, nenalezeno.*

*ABSTRACT:*

*Metoda POST vytvoří nového studenta v tabulce, ale se špatným statusem.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: {"id": 1569,"firstName": "Radek","lastName": "Bolševník","email": "bols@test.cz","age": 42}*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, jestli se student s ID 1569 vytvořil.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, Student byl v tabulce vytvořen.*

*Očekáváný výsledek (Expected result):*

*Status 201: Created, Student byl v tabulce vytvořen.*

*ABSTRACT:*

*Metoda POST vytvoří studenta s již existujícím ID a emailem a původního studenta přepíše.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: {"id":290,"firstName":"Miru","lastName":"JETOTAM","email":"jet@otam.cz","age":39}*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, co se stalo s již existujícím studentem s ID 290*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, nový student byl vytvořen pod ID stávajícího studenta a stávající student byl vymazán.*

*Očekáváný výsledek (Expected result):*

*Status 409 : Conflict, není možné vytvářet studenta pod již existujícím ID nebo s existujícím emailem.*

*ABSTRACT:*

*Metoda POST nevytvoří studenta s příliš dlouhým ID nebo s ID složeného ze speciálních či numerických znaků nebo písmenné ID nebo prázdnou hodnotu.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: {"id":50000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000055555555555555555555555555555555555555555555555555555555555555555555555555555555555555555555555,"firstName": "Láďa","lastName":"Průša","email","prusa@test.cz","age": 58} nebo místo ID použiji speciální znaky "id": @<>, nebo numerické znaky  "id": +-\*/= nebo písmena znaky  "id": LKM, nebo diakritická znaménka "id": ´ˇ.*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji v tabulce, jestli se student opravdu nevytvořil*

*Skutečný výsledek (Actual result):*

*Status 400: Bad request, ve všech případech program ohlásil špatný požadavek a student se v tabulce nevytvořil.*

*Očekávaný výsledek (Expected result):*

*Status 400: Bad request, ve všech případech program ohlásil špatný požadavek a student se v tabulce nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta i přesto, že jako firstName zadáme číselné hodnoty nebo hodnoty složené ze speciálních znaků nebo numerická znaménka nebo diakritická znaménka nebo nesmyslné jméno nebo velmi dlouhé hodnoty.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže, z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: { "id": 1588, "firstName": "5894", "lastName": "Novaková", "email": "postapost.cz", "age": 31}, nebo jako firstName"@@@@",nebo "firstName": "-\*/+", nebo "firstName": "?!´" nebo firstName": "ules" nebo "firstName": "uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu"*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student opravdu nevytvořil*

*Skutečný výsledek (Actual result):*

***X****Status 200:OK, student byl vytvořen v každém případě.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, nebyly vyplněny všechny povinné hodnoty a student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta i přesto, že jako firstName zadáme prázdnou hodnotu nebo mezeru.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže, z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: { "id": 1588, "firstName": " ", "lastName": "Novaková", "email": "postapost.cz", "age": 31}, nebo "firstName": ""*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student opravdu nevytvořil*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta, pokud jako lastName zadáme prázdnou hodnotu nebo mezeru.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže, z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: { "id": 1588, "firstName": " Magdaléna", "lastName": " ", "email": "postapost.cz", "age": 31}, nebo "lastName": ""*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student opravdu nevytvořil*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta i přesto, že jako lastName zadáme číselné hodnoty nebo hodnoty složené ze speciálních znaků nebo numerická znaménka nebo diakritická znaménka nebo nesmyslné jméno nebo prázdnou hodnotu nebo velmi dlouhé hodnoty.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: { "id": 1588, "firstName": "Jana", "lastName": " ", "email": "postapost.cz", "age": 31}, nebo jako lastName"@@@",nebo "lastName": "\*-+", nebo "lastName": "?!:" nebo "lastName": "hchkrdtn" nebo "uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu"*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student vytvořil či ne*

*Skutečný výsledek (Actual result):*

***X****Status 200:OK, student byl vytvořen v každém případě.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, student se nevytvořil.*

*ABSTRACT:*

*Metoda POST nevytvoří studenta, pokud jako age zadáme hodnoty složené ze speciálních znaků nebo numerická znaménka nebo diakritická znaménka nebo nesmyslný údaj nebo prázdnou hodnotu.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: { "id": 1588, "firstName": "Jana", "lastName": "5482", "email": "postapost.cz", "age": @@}, nebo jako "age": -+, nebo "age": ´ˇ nebo "age": 245 nebo prázdnou hodnotu.*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student v opravdu nevytvořil*

*Skutečný výsledek (Actual result):*

*Status 400: Bad request, student se v žádném z případů nevytvořil.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, student se v žádném z případů nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta při zadání emailu ve špatném formátu.*

*Steps to reproduce(kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: "id": 1530,"firstName": "Láďa","lastName": "Průša", "email": "prusa@", "age": 58} nebo "email": "" nebo "email": " " nebo "email": "uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu@dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd.cz"*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, zda se student v tabulce vytvořil či ne*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, program ohlásil chybu a student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta při zadání emailu s mezerou nebo prázdnou hodnotou.*

*Steps to reproduce(kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: "id": 1530,"firstName": "Jana","lastName": "Novaková", "email": "", "age": 31} nebo "email": " ".*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, zda se student vytvořil či ne*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil bez emailu.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, program ohlásil chybu a student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta při zadání již existujícího emailu.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu nového studenta, ale s již existujícím emailem: {"id":1763,"firstName":"Vivi","lastName":"VYHŘEZLÁ","email":"jjohn34@yourmail.com","age":25}*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *V tabulce zkontroluji, zda se student opravdu vytvořil či ne.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil bez emailu.*

*Očekáváný výsledek (Expected result):*

*Status 409: Conflict, program ohlásil chybu a student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta, s nevyplněnými údaji u firstName, lastName.*

*Steps to reproduce(kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: {"id":1584,"firstName":"","lastName":"","email":"","age":58}*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student vytvořil či ne.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil i bez vyplnění všech požadovaných hodnot.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad regest, Program ohlásí, že chybí vyplnit některé hodnoty a student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta pod jiným ID, i přestože v tabulce již existuje student se stejnými ostatními údaji.*

*Steps to reproduce(kroky):*

1. *Přes vyvolání studenta metodou GET zkopíruji existujícího studenta.*
2. *Na začátku příkazového řádku vyberu metodu POST*
3. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
4. *Pod příkazovým řádkem ze záložky zvolím BODY*
5. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
6. *Do editoru vypíšu zkopírovaného studenta: {"id":613,"firstName":"josef","lastName": "KONOPA","email":"postapost.cz","age":31}*
7. *Vpravo nahoře zmáčknu SEND*
8. *Zkontroluji status*
9. *Ve WorkBench aktualizuji tabulku*
10. *Zkontroluji, zda se student vytvořil duálně či ne.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se duálně vytvořil se stejnými údaji.*

*Očekáváný výsledek (Expected result):*

*Status 409: Conflict, Student se duálně nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta i s nesmyslným věkem (-7, 0, 1, 160) nebo s příliš dlouhou hodnotou.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: {"id":1772,"firstName":"Test","lastName":"OHLÁ","email":"Ohl5@yourmail.com","age":160} nebo "age":0 nebo "age":1 nebo "age":-7 nebo "age": 1014.*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, zda se student vytvořil či ne.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil v každém případě.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad reguest, student se nevytvořil z důvodu špatně zadaného požadavku.*

*ABSTRACT:*

*Metoda POST správně zaokrouhlí desetinné číslo na číslo celé u sloupce s ID a age.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: "id": 1779.9, "firstName": "Mima","lastName": "Zvenuše", "email": "Mima@Vesmir.com", "age": 50,5}*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, zda se student vytvořil či ne, a jestli se des. číslo převedlo na číslo celé.*

*Skutečný výsledek (Actual result):*

*Status 200: OK, student se vytvořil a čísla se zaokrouhlila.*

*Očekáváný výsledek (Expected result):*

*Status 200: OK, student se vytvořil a čísla se zaokrouhlila.*

*ABSTRACT:*

*Postman správně vytvoří studenta přes metodu POST, vytáhne data z tabulky na Workbench za pomocí metody GET a studenta zase smaže pomocí metody DELETE.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Před příkazovým řádkem zvolím metodu POST.*
3. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
4. *Pod příkazovým řádkem ze záložky zvolím BODY*
5. *O řádek níže, z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
6. *Do editoru vypíšu zkopírovaného studenta: {"id":1778,"firstName":"Mima","lastName":"ZVENUŠE","email":"Mima@Vesmir.com","age":50}*
7. *Vpravo nahoře zmáčknu SEND*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student vytvořil a údaje se shodují*
10. *Metodu POST změním na metodu GET a zadám ID 1178 toho samého studenta*
11. *Vpravo nahoře zmáčku SEND*
12. *Zkontroluji, jestli se data v Postman shodují s údaji z tabulky na WorkBench .*
13. *Metodu GET změním na metodu DELETE*
14. *Vpravo nahoře zmáčkun SEND*
15. *Zkontroluji jestli se ve WorkBench student s ID 1178 smazal.*

*Skutečný výsledek (Actual result):*

*Status 200: OK, všechny operace proběhli úspěšně.*

*Očekáváný výsledek (Expected result):*

*Status 200: OK, všechny operace proběhli úspěšně.¨*

*ABSTRACT:*

*Metoda POST vypíše výsledek v přijatelném čase.*

*Steps to reproduce(kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: {"id":613,"firstName":"josef","lastName": "KONOPA","email":"postapost.cz","age":31}*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student správně vytvořil.*
10. *Vedle statusu zkontroluji čas vypsání*

*Skutečný výsledek (Actual result):*

*Status 200: OK, student se vytvořil v přijatelném čase 192 ms.*

*Očekáváný výsledek (Expected result):*

*Status 200: OK, student se vytvořil v přijatelném čase 192 ms.*

*ABSTRACT:*

*Metoda GET vypíše výsledek v přijatelném čase.*

*Steps to reproduce(kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Nakonec URL adresy napíšu ID existujícího studenta 477*
4. *Vpravo nahoře zmáčknu SEND*
5. *Zkontroluji status*
6. *Ve WorkBench aktualizuji tabulku*
7. *Zkontroluji, zda se údaje shodují*
8. *Vedle statusu zkontroluji čas vypsání*

*Skutečný výsledek (Actual result):*

*Status 200: OK, student byl vypsán v přijatelném čase 135 ms.*

*Očekáváný výsledek (Expected result):*

*Status 200: OK, student byl vypsán v přijatelném čase 135 ms.*

*ABSTRACT:*

*Metoda DELETE smaže studenta v přijatelném čase.*

*Steps to reproduce(kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Nakonec URL adresy napíšu ID existujícího studenta 1761*
4. *Vpravo nahoře zmáčknu SEND*
5. *Zkontroluji status*
6. *Ve WorkBench aktualizuji tabulku*
7. *Zkontroluji, zda se údaje shodují*
8. *Vedle statusu zkontroluji čas vypsání*

*Skutečný výsledek (Actual result):*

*Status 200: OK, student byl smazán v přijatelném čase 102 ms.*

*Očekáváný výsledek (Expected result):*

*Status 200: OK, student byl smazán v přijatelném čase 102 ms.*

*ABSTRACT:*

*Metoda PUT neaktualizuje údaje z důvodu nepovoleného přístupu k této metodě, ale vypíše nevhodný status.*

*Steps to reproduce(kroky):*

1. *Otevřu SQL Workbench, aktualizuji tabulku a zjistím, že student 1603 nemá uložený lastName a email.*
2. *Přejdu zpět do Postman.*
3. *Na začátku příkazového řádku vyberu metodu PUT*
4. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
5. *Za URL adresu napíšu číslo existujícího studenta 1603*
6. *Pod příkazovým řádkem ze záložky zvolím BODY*
7. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
8. *Do editoru vypíšu studenta 1603 a doplním chybějící údaje: "id": 1603,"firstName": "Jana", "lastName": "Junova","email": "jana@gmail.com", "age": 48}*
9. *Vpravo nahoře zmáčknu SEND*
10. *Ve WorkBench aktualizuji tabulku*
11. *Zkontroluji, zda se student opravdu neaktualizoval*

*Skutečný výsledek (Actual result):*

***X****Status 405: Method not allowed, Požadavek byl zamítnut a student se neaktualizoval.*

*Očekáváný výsledek (Expected result):*

*Status 403: Forbidden, Požadavek byl zamítnut a student se neaktualizoval.*

*ABSTRACT:*

*Toto Rest API nepodporuje stránkování ani filtrování.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu GET*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání plus příkaz (*[*http://108.143.193.45:8080/api/v1/students/?firstName=Radek*](http://108.143.193.45:8080/api/v1/students/?firstName=Radek) *nebo* [*http://108.143.193.45:8080/api/v1/students/?page=1&limit=10*](http://108.143.193.45:8080/api/v1/students/?page=1&limit=10) *)*
3. *Vpravo nahoře zmáčknu SEND*
4. *Ve výpisu ve spodní části Postman zkontroluji výpisy studentů a jestli jsou vypsáni pouze studenti s firstName Radek nebo ve 2. případě 10 studentů z první stránky.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, byli však vypsáni všichni studenti v obou případech.*

*Očekáváný výsledek (Expected result):*

*Status 501: Not implemented, nepodporováno.*

**BUG REPORT**

**Na základě provedených scénářů jsem objevila uvedené chyby aplikace.**

*ABSTRACT:*

*Metoda GET nevypíše neexistujícího studenta, ale vypíše špatný status.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu GET*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání.*
3. *Na konec url adresy zadám neexistujícího studenta s Id 1*
4. *Vpravo nahoře zmáčknu SEND*
5. *Porovnám výsledky s tabulkou na workbench.*

*Skutečný výsledek (Actual result):*

***X****Status 500: Internal error, student nebyl nalezen.*

*Očekáváný výsledek (Expected result):*

*Status 404: Not found, nenalezeno.*

*ABSTRACT:*

*Metoda DELETE správně vypíše chybu při mazání neexistujícího studenta, ale zvolí špatný status.*

*Steps to reproduce:*

1. *Na začátku příkazového řádku vyberu metodu DELETE*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání.*
3. *Za lomítko, na konec URL adresy, zadám (dle tabulky) neexistujícího studenta s Id 317*
4. *Porovnám výsledky s tabulkou na WorkBench, zda nedošlo ke změnám.*

*Skutečný výsledek (Actual result):*

***X****Status 500: Internal server error, student nebyl nalezen.*

*Očekáváný výsledek (Expected result):*

*Status 404: Not found, nenalezeno.*

*ABSTRACT:*

*Metoda DELETE vypíše chybu při mazání již smazaného studenta, ale se špatným statusem.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu DELETE*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání.*
3. *Za lomítko, na konec url adresy, zadám (dle tabulky) neexistujícího studenta s Id 1777*
4. *Porovnám výsledky s tabulkou na WorkBench, zda nedošlo ke změnám.*

*Skutečný výsledek (Actual result):*

***X****Status 500:Internal server error, student nebyl nalezen.*

*Očekáváný výsledek (Expected result):*

*Status 410: Gone, nenalezeno.*

*ABSTRACT:*

*Metoda POST vytvoří nového studenta v tabulce, ale se špatným statusem.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: {"id": 1569,"firstName": "Radek","lastName": "Bolševník","email": "bols@test.cz","age": 42}*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, jestli se student s ID 1569 vytvořil.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, Student byl v tabulce vytvořen.*

*Očekáváný výsledek (Expected result):*

*Status 201: Created, Student byl v tabulce vytvořen.*

*ABSTRACT:*

*Metoda POST vytvoří studenta s již existujícím ID a emailem a původního studenta přepíše.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: {"id":290,"firstName":"Miru","lastName":"JETOTAM","email":"jet@otam.cz","age":39}*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, co se stalo s již existujícím studentem s ID 290*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, nový student byl vytvořen pod ID stávajícího studenta a stávající student byl vymazán.*

*Očekáváný výsledek (Expected result):*

*Status 409 : Conflict, není možné vytvářet studenta pod již existujícím ID nebo s existujícím emailem.*

*ABSTRACT:*

*Metoda POST vytvoří studenta i přesto, že jako firstName zadáme číselné hodnoty nebo hodnoty složené ze speciálních znaků nebo numerická znaménka nebo diakritická znaménka nebo nesmyslné jméno nebo velmi dlouhé hodnoty.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže, z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: { "id": 1588, "firstName": "5894", "lastName": "Novaková", "email": "postapost.cz", "age": 31}, nebo jako firstName"@@@@",nebo "firstName": "-\*/+", nebo "firstName": "?!´" nebo firstName": "ules" nebo "firstName": "uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu"*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student opravdu nevytvořil*

*Skutečný výsledek (Actual result):*

***X****Status 200:OK, student byl vytvořen v každém případě.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, nebyly vyplněny všechny povinné hodnoty a student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta i přesto, že jako firstName zadáme prázdnou hodnotu nebo mezeru.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže, z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: { "id": 1588, "firstName": " ", "lastName": "Novaková", "email": "postapost.cz", "age": 31}, nebo "firstName": ""*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student opravdu nevytvořil*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta, pokud jako lastName zadáme prázdnou hodnotu nebo mezeru.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže, z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: { "id": 1588, "firstName": " Magdaléna", "lastName": " ", "email": "postapost.cz", "age": 31}, nebo "lastName": ""*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student opravdu nevytvořil*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta i přesto, že jako lastName zadáme číselné hodnoty nebo hodnoty složené ze speciálních znaků nebo numerická znaménka nebo diakritická znaménka nebo nesmyslné jméno nebo prázdnou hodnotu nebo velmi dlouhé hodnoty.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: { "id": 1588, "firstName": "Jana", "lastName": " ", "email": "postapost.cz", "age": 31}, nebo jako lastName"@@@",nebo "lastName": "\*-+", nebo "lastName": "?!:" nebo "lastName": "hchkrdtn" nebo "uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu"*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student vytvořil či ne*

*Skutečný výsledek (Actual result):*

***X****Status 200:OK, student byl vytvořen v každém případě.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta při zadání emailu ve špatném formátu.*

*Steps to reproduce(kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: "id": 1530,"firstName": "Láďa","lastName": "Průša", "email": "prusa@", "age": 58} nebo "email": "" nebo "email": " " nebo "email": "uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu@dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd.cz"*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, zda se student v tabulce vytvořil či ne*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, program ohlásil chybu a student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta při zadání emailu s mezerou nebo prázdnou hodnotou.*

*Steps to reproduce(kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: "id": 1530,"firstName": "Jana","lastName": "Novaková", "email": "", "age": 31} nebo "email": " ".*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, zda se student vytvořil či ne*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil bez emailu.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad request, program ohlásil chybu a student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta při zadání již existujícího emailu.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu nového studenta, ale s již existujícím emailem: {"id":1763,"firstName":"Vivi","lastName":"VYHŘEZLÁ","email":"jjohn34@yourmail.com","age":25}*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *V tabulce zkontroluji, zda se student opravdu vytvořil či ne.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil bez emailu.*

*Očekáváný výsledek (Expected result):*

*Status 409: Conflict, program ohlásil chybu a student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta, s nevyplněnými údaji u firstName, lastName.*

*Steps to reproduce(kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu: {"id":1584,"firstName":"","lastName":"","email":"","age":58}*
6. *Vpravo nahoře zmáčknu SEND*
7. *Zkontroluji status*
8. *Ve WorkBench aktualizuji tabulku*
9. *Zkontroluji, zda se student vytvořil či ne.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil i bez vyplnění všech požadovaných hodnot.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad regest, Program ohlásí, že chybí vyplnit některé hodnoty a student se nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta pod jiným ID, i přestože v tabulce již existuje student se stejnými ostatními údaji.*

*Steps to reproduce(kroky):*

1. *Přes vyvolání studenta metodou GET zkopíruji existujícího studenta.*
2. *Na začátku příkazového řádku vyberu metodu POST*
3. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
4. *Pod příkazovým řádkem ze záložky zvolím BODY*
5. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
6. *Do editoru vypíšu zkopírovaného studenta: {"id":613,"firstName":"josef","lastName": "KONOPA","email":"postapost.cz","age":31}*
7. *Vpravo nahoře zmáčknu SEND*
8. *Zkontroluji status*
9. *Ve WorkBench aktualizuji tabulku*
10. *Zkontroluji, zda se student vytvořil duálně či ne.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se duálně vytvořil se stejnými údaji.*

*Očekáváný výsledek (Expected result):*

*Status 409: Conflict, Student se duálně nevytvořil.*

*ABSTRACT:*

*Metoda POST vytvoří studenta i s nesmyslným věkem (-7, 0, 1, 160) nebo s příliš dlouhou hodnotou.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu POST*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
3. *Pod příkazovým řádkem ze záložky zvolím BODY*
4. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
5. *Do editoru vypíšu zkopírovaného studenta: {"id":1772,"firstName":"Test","lastName":"OHLÁ","email":"Ohl5@yourmail.com","age":160} nebo "age":0 nebo "age":1 nebo "age":-7 nebo "age": 1014.*
6. *Vpravo nahoře zmáčknu SEND*
7. *Ve WorkBench aktualizuji tabulku*
8. *Zkontroluji, zda se student vytvořil či ne.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, student se vytvořil v každém případě.*

*Očekáváný výsledek (Expected result):*

*Status 400: Bad reguest, student se nevytvořil z důvodu špatně zadaného požadavku.*

*ABSTRACT:*

*Metoda PUT neaktualizuje údaje z důvodu nepovoleného přístupu k této metodě, ale vypíše nevhodný status.*

*Steps to reproduce(kroky):*

1. *Otevřu SQL Workbench, aktualizuji tabulku a zjistím, že student 1603 nemá uložený lastName a email.*
2. *Přejdu zpět do Postman.*
3. *Na začátku příkazového řádku vyberu metodu PUT*
4. *Do příkazového řádku vložím URL adresu z ENGETO zadání*
5. *Za URL adresu napíšu číslo existujícího studenta 1603*
6. *Pod příkazovým řádkem ze záložky zvolím BODY*
7. *O řádek níže z výběru zvolím RAW a na konci řádku z výběru zvolím Json*
8. *Do editoru vypíšu studenta 1603 a doplním chybějící údaje: "id": 1603,"firstName": "Jana", "lastName": "Junova","email": "jana@gmail.com", "age": 48}*
9. *Vpravo nahoře zmáčknu SEND*
10. *Ve WorkBench aktualizuji tabulku*
11. *Zkontroluji, zda se student opravdu neaktualizoval*

*Skutečný výsledek (Actual result):*

***X****Status 405: Method not allowed, Požadavek byl zamítnut a student se neaktualizoval.*

*Očekáváný výsledek (Expected result):*

*Status 403: Forbidden, Požadavek byl zamítnut a student se neaktualizoval.*

*ABSTRACT:*

*Toto Rest API nepodporuje stránkování ani filtrování.*

*Steps to reproduce (kroky):*

1. *Na začátku příkazového řádku vyberu metodu GET*
2. *Do příkazového řádku vložím URL adresu z ENGETO zadání plus příkaz (*[*http://108.143.193.45:8080/api/v1/students/?firstName=Radek*](http://108.143.193.45:8080/api/v1/students/?firstName=Radek) *nebo* [*http://108.143.193.45:8080/api/v1/students/?page=1&limit=10*](http://108.143.193.45:8080/api/v1/students/?page=1&limit=10) *)*
3. *Vpravo nahoře zmáčknu SEND*
4. *Ve výpisu ve spodní části Postman zkontroluji výpisy studentů a jestli jsou vypsáni pouze studenti s firstName Radek nebo ve 2. případě 10 studentů z první stránky.*

*Skutečný výsledek (Actual result):*

***X****Status 200: OK, byli však vypsáni všichni studenti v obou případech.*

*Očekáváný výsledek (Expected result):*

*Status 501: Not implemented, nepodporováno.*