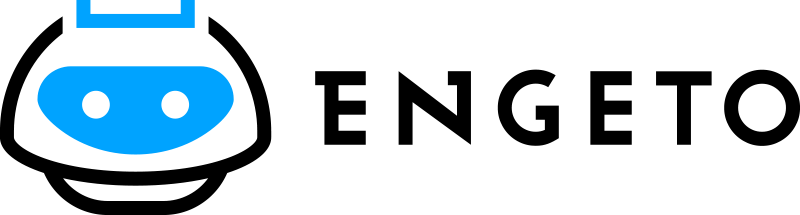
FINÁLNÍ PROJEKT   
 č.1



Autor: Gabriela Řempělová  
Datum:

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

Čílem 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..

POST - uložit se musí všechny atributy a vše lowerCase.

## Přístupové údaje:

|  |  |
| --- | --- |
| Databáze | database: qa\_demo Host: aws.connect.psdb.cloud Username: 68q0llcbuvyijdt5mzq2 Password: pscale\_pw\_zxfenAXfSYx9loe9PCS1snuGKrFzw2J84BMIVZI96o |
| 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

|  |
| --- |
| **GET:** |

1. GET ⇒ Abstrakt: test existujícího studenta id: 219

**Hodnoty v DTB:**  id: 219

age: 34

email: [jjohn34@yourmail.com](mailto:jjohn34@yourmail.com)

firstName: “ “

lastName: “JOSHUA”

*Pozn. prázdná hodnota v poli firstName*

|  |  |
| --- | --- |
| *Kroky:* | *Očekávaný výsledek:* |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 1. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se bez problému otevře nový požadavek |
| 1. pod text “Untitled Request” vyberu z nabídky “GET” a do políčka napravo vložit “<http://108.143.193.45:8080/api/v1/students/219>” a kliknout vpravo na tlačítko “SEND” | 3. po zvolení GET, vložení uvedeného URL a kliknutí na SEND, se zobrazí dole v části “BODY” příslušné data pro studenta id 219 |
| 1. Dole v sekci “BODY” na pravé straně se zobrazí status code “200 OK” |  |
| 1. Pod sekci “BODY” se zobrazí příslušné údaje k id 219 ve formátu **JSON**:   {  "id": 219,  "firstName": "",  "lastName": "JOSHUA",  "email": "jjohn34@yourmail.com",  "age": 34  } |  |
| 1. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem |
| 1. Do části Query doplnit:   SELECT  \*  FROM student  WHERE id=219  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Po spuštění příkazu se dole zobrazí v “Result Grid” příslušné údaje ke studentovi id 219:    id: 219  age: 34  email: “[jjohn34@yourmail.com](mailto:jjohn34@yourmail.com)”  first\_name: “ “  last\_name: “JOSHUA” |
| 1. Porovnat údaje ke studentovi id 219 z Postamen a MySQL WorkBench - zda jsou všechny shodné a zobrazily se správně:  * DTB id: **219** = Postmane id: **219** * DTB age: **34** = Postmane age: **34** * DTB email: [jjohn34@yourmail.com](mailto:jjohn34@yourmail.com) = Postamen email: [jjohn34@yourmail.com](mailto:jjohn34@yourmail.com) * DTB first\_name: “ “ = Postmane firstName: “ “ * DTB last\_name: “**JOSHUA**” = Postamen lastName: “**JOSHUA**” | Všechny údaje z DTB se shodují s údaji v Postamen a zobrazují se kompletně:   * DTB id: **219** = Postmane id: **219** * DTB age: **34** = Postmane age: **34** * DTB email: [jjohn34@yourmail.com](mailto:jjohn34@yourmail.com) = Postamen email: [jjohn34@yourmail.com](mailto:jjohn34@yourmail.com) * DTB first\_name: “ “ = Postmane firstName: “ “ * DTB last\_name: “**JOSHUA**” = Postamen lastName: “**JOSHUA**” |

1. GET ⇒ Abstrakt: test existujícího studenta **id: 307**

**Hodnoty v DTB:**  id: 307

age: 89

email: ýá+ěčzč51

firstName: “-86 “

lastName: “\*+6”

*POZN.: kombinace znaků a čísel v primárně textových polích*

|  |  |
| --- | --- |
| *Kroky:* | *Očekávaný výsledek:* |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 1. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se bez problému otevře nový požadavek |
| 1. pod text “Untitled Request” vybrat z nabídky “GET” a do políčka napravo zkopirovat tento odkaz “http://108.143.193.45:8080/api/v1/students/307” a kliknout vpravo na tlačítko “SEND” | 3. po zvolení GET, vložení uvedeného URL a kliknutí na SEND se zobrazí dole v části “BODY” příslušné data pro studenta id 307 |
| 1. Dole v sekci “BODY” na pravé straně se zobrazí status code “200 OK” |  |
| 1. Pod sekci “BODY” se zobrazí příslušné údaje k id 307 ve formátu **JSON**:   {  "id": 307,  "firstName": "-86",  "lastName": "\*+6",  "email": "ýá+ěčzč51",  "age": 89  } |  |
| 1. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem |
| 1. Do části Query doplnit:   SELECT  \*  FROM student  WHERE id=307  ;  a spustit požadavek prostřednictvím CTRL+enter | 7. Po spuštění příkazu se dole zobrazí v “Result Grid” příslušné údaje ke studentovi id 307:    id: 307  age: 89  email: “ýá+ěčzč51”  first\_name: “-86 “  last\_name: “\*+6” |
| 1. Porovnat údaje ke studentovi id 307 z Postamen a MySQL WorkBench - zda jsou všechny shodné a zobrazily se správně:  * DTB id: **307** * DTB age: **89** * DTB email: ýá+ěčzč51 * DTB first\_name: “**-86**” * DTB last\_name: “**\*+6**” | 8. Všechny údaje z DTB se shodují s údaji v Postamen a zobrazují se kompletně:   * DTB id: **307 =** Postman id: **307** * DTB age: **89** = Postman age: **89** * DTB email: ýá+ěčzč51 = Postman email: "ýá+ěčzč51" * DTB first\_name: “**-86**” = firstName: **"-86"** * DTB last\_name: “**\*+6**” = lastName: **"\*+6"** |

1. GET ⇒ Abstrakt: test neexistujícího studenta **id: 10**

*POZN.: student s tímto id neexistuje*

|  |  |
| --- | --- |
| *Kroky:* | *Očekávaný výsledek:* |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 1. po kliknutí na “NEW” a zvolení “HTTP” se bez problému otevře nový požadavek |
| 3. pod text “Untitled Request” vybrat z nabídky “GET” a do políčka napravo vložit tento odkaz - “http://108.143.193.45:8080/api/v1/students/10” a kliknout vpravo na tlačítko “SEND” | 1. lze vybrat GET, vložit uvedeného URL a kliknout na SEND |
| 1. V pravo v sekci “BODY” se objeví chybová hláška | 4. zobrazí se chybová hláška 404 - neexistující student resp. student s tímto ID neexistuje |
| 1. V sekci BODY se zobrazí potvrzení chybové hlášky | 5. v sekci BODY se zobrazí potvrzení chybové hlášky s upřesněním, že student s tímto ID neexistuje |
| 1. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem |
| 1. Do části Query doplnit:   SELECT  \*  FROM student  WHERE id=10  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Po spuštění příkazu se dole zobrazí v “Result Grid” prázdné informace, protože student s ID 10 neexistuje v DTB a zároveň v části Action Output se zobrazí error “hláška” - student s tímto ID nenalezen. |

1. GET ⇒ Abstrakt: test neexistujícího studenta id: **xy14**

*POZN.: neexistující ID studenta, ID v kombinaci znaků a čísel*

|  |  |
| --- | --- |
| *Kroky:* | *Očekávaný výsledek:* |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 1. po kliknutí na “NEW” a zvolení “HTTP” se bez problému otevře nový požadavek |
| 3. pod text “Untitled Request” vybrat z nabídky “GET” a do políčka napravo zkopírovat tento odkaz - “http://108.143.193.45:8080/api/v1/students/xy14” a kliknu vpravo na tlačítko “SEND” | 1. lze zvolit GET, vložit uvedené URL a kliknout na tlačítko SEND |
| 4. V dolní části se objeví v pravo chybová hláška - chybový status code - 404 - neexistující student | 1. Zobrazí se chybová hláška - status code 404 - neexistující student resp. student s tímto ID neexistuje |
| 5. Dole v sekci “BODY” se objeví doplňující chybová hláška | 1. v sekci BODY se zobrazí potvrzení chybové hlášky s upřesněním, že student s tímto ID neexistuje |
| 1. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=xy14  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Po spuštění příkazu se dole zobrazí v “Result Grid” nezobrazí žádné informace, protože student s ID xy14 neexistuje a zároveň v části Action Output se zobrazí error “hláška” |

|  |
| --- |
| **POST:** |

(požadavky - vše ukládat malými písmeny + povinné všechny hodnoty v DTB)

1. POST ⇒ Abstrakt: založit studenta se všemi údaji “správně”- všechny atributy vyplněny ve správném formátu

**Hodnoty:**  id:

age: 39

email: pavel.nevimkdo@gmail.com

firstName: “ Pavel“

lastName: “Nevimkdo”

|  |  |
| --- | --- |
| Kroky: | Očekávaný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 1. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek |
| 1. vybrat z nabídky POST (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)”, | 3. lze vybrat variantu POST a vložit uvedené URL |
| 1. Dole v části BODY, zakliknout možnost “raw”, rozkliknou možnost “TEXT” a zvolit variantu “JSON” | 4. v části BODY lze zakliknout možnost “raw” a vybrat variantu “JSON” v části “TEXT” |
| 1. v nově otevřeném okně po zakliknutí “raw” doplnit tento text:   {  "firstName": "Pavel",  "lastName": "Nevimkdo",  "email": "pavel.nevimkdo@gmail.com",  "age": 39  }  a kliknu na SEND | 5. v pořádku lze zapsat uvedené údaje a kliknout na “SEND” |
| 1. zobrazí se status code 200 | 6. zobrazí se status code 200 |
| 1. V spodní části BODY se zobrazí doplněné údaje ve formátu JSON a přidělí se ID studenta | 7. v spodní části se zobrazí veškeré zadané údaje a navíc se doplní automaticky ID studenta |
| 1. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 8. Lze otevřít aplikaci Workbench a připojit se na uvedenou DTB |
| 1. do části Query doplnit tento text:   SELECT  \*  FROM student  WHERE id=972  ;  potvrdit CTRL+ENTER | 9. lze doplnit uvedené údaje a spustit požadavek CTRL+ENTER |
| 1. Porovnat údaje ke studentovi id 972 z Postamen a MySQL WorkBench - vše se shoduje, všechny atributy jsou zapsány správně - malými písmeny a jsou vyplněné všechny atributy: | 10. Všechny atributy k vytvořenému studentovi v DTB se shodují s údaji v Postman a zobrazují se kompletně, malými písmeny:   * DTB id: **972=** Postaman id: **972** * DTB age: **39** = Postman age: **39** * DTB email: **pavel.nevimkdo@gmail.com** = Postman email: **pavel.nevimkdo@gmail.com** * DTB first\_name: “**pavel**” = Postman firstName: **"Pavel"** * DTB last\_name: “**\*nevimkdo**” = lastName: **"\*Nevimkdo"** |

1. POST ⇒ Abstrakt: založit studenta s chybějícími údaji ve jméně:

**Hodnoty:**  id:

age: 21

email: jsem.zde.kamo@seznam.cz

firstName: “ “

lastName: “Kamos”

|  |  |
| --- | --- |
| Kroky: | Očekávaný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek |
| 3. vybrat z nabídky POST (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)”, | 3. lze vybrat variantu POST a vložit uvedené URL |
| 4. Dole v části BODY, zakliknout možnost “raw”, rozkliknout možnost “TEXT” a zvolit variantu “JSON” | 4. v části BODY lze zakliknout možnost “raw” a vybrat variantu “JSON” v části “TEXT” |
| 5. v nově otevřeném okně po zakliknutí “raw” doplnit tento text:  {  "firstName": " ",  "lastName": "Kamos",  "email": "jsem.zde.kamo@seznam.cz",  "age": 21  }  a kliknu na SEND | 5. v pořádku se zapíšou uvedené údaje a lze kliknout na “SEND” |
| 6. zobrazí se status code 400 - který upozorní, že nejsou vyplněné všechny atrbuty | 6. zobrazí se error status code 400 s informací, že nejsou vyplněné všechny údaje |
| 7. V spodní části BODY se zobrazí doplnění chybové hlášky, že není vyplněn atribut “firstName” | 7. v spodní části se zobrazí informace, že není vyplněn atribut “firstName” a je nutné jej doplnit - “Vyplňte položku jméno.” |
| 8. Postman studenta neuloží, dokud nebudou vyplněné všechny atributy. | 8. Postman tohoto studenta neuloží, dokud nebudou vyplněné všechny atributy včetně momentálně prázdného pole “firstName”. Studentovi s prázdným atributem “firstName” není přiděleno ID. |

1. POST ⇒ Abstrakt: založit studenta se špatným formátem v e-mailu a špatným formátem věku:

**Hodnoty:**  id:

age: -5

email: ahojkamocentrum.cz.com

firstName: “Jiřík “

lastName: “Kocourek”

|  |  |
| --- | --- |
| Kroky: | Očekávaný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek |
| 3. vybrat z nabídky POST (pod části Untitled Request) a vložit do kolonky vedle toto URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)”, | 3. lze vybrat variantu POST a vložit uvedené URL |
| 4. Dole v části BODY, zakliknout možnost “raw”, rozkliknout možnost “TEXT” a zvolit variantu “JSON” | 4. v části BODY lze zakliknout možnost “raw” a vybrat variantu “JSON” v části “TEXT” |
| 5. v nově otevřeném okně po zakliknutí “raw” doplnit tento text:  {  "firstName": "Jiřík ",  "lastName": "Kocourek",  "email": "ahojkamocentrum.cz.com",  "age": -5  }  a kliknu na SEND | 5. v pořádku se zapíšou uvedené uvedené údaje a kliknout na “SEND” |
| 6. zobrazí se status code 400 - který upozorní, že atribut “age” a “email” mají špatný formát. | 6. zobrazí se error status code 400 s informací, že atribut “age” a “email” jsou ve špatném formátu. |
| 7. V spodní části BODY se zobrazí doplnění chybové hlášky, že atributy “age” a “email” mají špatný formát a je nutné je vyplnit správně. | 7. v spodní části se zobrazí informace, že není atribut “age” a “email” jsou ve špatném formátu. Je nutné je vyplnit správně. |
| 8. Postman studenta neuloží, dokud nebudou vyplněné všechny atributy ve správném formátu. | 8. Postman tohoto studenta neuloží, dokud nebudou vyplněné správně všechny atributy včetně momentálně špatně zadaného atributu “age” = -5(záporná hodnota) a atributu “email” - chybí @. |

1. POST ⇒ Abstrakt: založit totožného studenta jako v POST abstrakt č. 1 - všechny hodnoty totožné (kontrola duplicit)

(Postman by měl upozornit, že student s těmito údaji je již založen - kontrola duplicitního uložení studentu dle dokumentace - pro tyto potřeby kontrola dle e-mailu)

**Hodnoty:**  id:

age: 39

email: pavel.nevimkdo@gmail.com

firstName: “ Pavel“

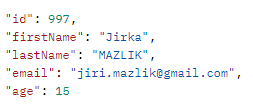
lastName: “Nevimkdo”

|  |  |
| --- | --- |
| Kroky: | Očekávaný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek |
| 3. vybrat z nabídky POST (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)”, | 3. lze vybrat variantu POST a vložit uvedené URL |
| 4. Dole v části BODY, zakliknout možnost “raw”, rozkliknout možnost “TEXT” a zvolit variantu “JSON” | 4. v části BODY lze zakliknout možnost “raw” a vybrat variantu “JSON” v části “TEXT” |
| 5. v nově otevřeném okně po zakliknutí “raw” doplnit tento text:  {  "firstName": "Pavel",  "lastName": "Nevimkdo",  "email": "pavel.nevimkdo@gmail.com",  "age": 39  }  a kliknu na SEND | 5. v pořádku se zapíšou uvedené uvedené údaje a kliknout na “SEND” |
| 6. zobrazí se status code 400 - který upozorní, že student s tímto email již existuje | 6. zobrazí se error status code 400 s informací, že student s tímto s tímto emailem existuje |
| 7. V spodní části BODY se zobrazí doplnění chybové hlášky, že studenta nelze uložit, protože student s tímto e-mailem již existuje. | 7. v spodní části se zobrazí informace, že student s těmito údaji je již založen. |
| 8. Postman studenta neuloží, hlídá duplicitní uložení studentů dle jedinečné hodnoty v atributu “email” - dle dokumentace. | 8. Postman studenta neuloží, hlídá duplicitní uložení studentů dle jedinečné hodnoty v atributu “email” - dle dokumentace. Pokud je student pod tímto e-mailem již založen, nelze pod totožným emailem zakládat dalšího studenta. Kontrola duplicit dle dokumentace (e-mail, příjmení, jméno….) |

# 

|  |
| --- |
| **DELETE:** |

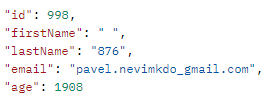
1. DELETE ⇒ Abstrakt: smazání studenta ID 997 (všechny atributy vyplněné)

****

|  |  |
| --- | --- |
| Kroky: | Očekávaný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek |
| 3. vybrat z nabídky DELETE (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)/997”,   * kliknout na tlačítko SEND | 3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND |
| 4. v části BODY se zobrazí vpravo status code 200 | 4. zobrazí se status code 200 |
| 5. v části BODY v dolní části nejsou vyplněné žádné údaj - student je smazán | 5. v části BODY nejsou žádné údaje - student je smazán |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=997  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Vložit uvedené příkaz a spustím prostřednictvím požadavek přes kombinace kláves CTRL+ enter. |
| 8. v Workbench se nezobrazí žádné údaje, protože student je smazán | 8. v Workbench se nezobrazí žádné údaje, protože student je smazán |

# 

1. DELETE ⇒ Abstrakt: smazání studenta ID 998 (prázdný atribut firstName)





|  |  |
| --- | --- |
| Kroky: | Očekávaný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek |
| 3. vybrat z nabídky DELETE (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)/998”,   * kliknout na tlačítko SEND | 3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND |
| 4. v části BODY se zobrazí vpravo status code 200 | 4. zobrazí se status code 200 |
| 5. v části BODY v dolní části nejsou vyplněné žádné údaj - studnt je smazán | 5. v části BODY nejsou žádné údaje - student je smazán |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=998  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Vložím uvedené příkaz a spustím jej prostřednictvím kombinace kláves CTRL+ enter. |
| 8. v Workbench se nezobrazí žádné údaje, protože student je smazán | 8. v Workbench se nezobrazí žádné údaje, protože student je smazán |

3. DELETE ⇒ Abstrakt: smazání studenta s neexistujícím ID - 1234 (id číselné hodnoty)

|  |  |
| --- | --- |
| Kroky: | Očekávaný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek |
| 3. vybrat z nabídky DELETE (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)/1234”,   * kliknout na tlačítko SEND | 3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND |
| 4. v části BODY se zobrazí vpravo error status code | 4. zobrazí se erro status code 404 - neexistující student resp. ID |
| 5. v části BODY v dolní části jsou doplňující informace k error statusu | 5. v části BODY v dolní části jsou doplňující informace k error statusu - “Student s tímto ID není uložen v databázi”, |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=1234  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Vložit uvedené příkaz a spustit jej prostřednictvím kombinace kláves CTRL+ enter. |
| 8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují a tudíž jej nelze smazat | 8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují a tudíž jej nelze smazat |

4. DELETE ⇒ Abstrakt: smazání studenta s neexistujícím ID s použitím speciálních znaků (podtržítko) ID 99\_8

|  |  |
| --- | --- |
| Kroky: | Očekávaný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek |
| 3. vybrat z nabídky DELETE (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)/99\_8”,   * kliknout na tlačítko SEND | 3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND |
| 4. v části BODY se zobrazí vpravo error status code | 4. zobrazí se error status code 400 - špatný formát položky ID |
| 5. v části BODY v dolní části jsou doplňující informace k error statusu | 5. v části BODY v dolní části jsou doplňující informace k error statusu - “Špatný formát položka ID, zadejte pouze čísla.”, |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=99\_8  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Vložit uvedené příkaz a spustit jej prostřednictvím kombinace kláves CTRL+ enter. |
| 8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují a tudíž jej nelze smazat - zobrazí se error status, zadané ID má špatný formát | 8. v Workbench se zobrazí error status - špatný formát položky ID |

5. DELETE ⇒ Abstrakt: smazání studenta s neexistujícím ID s použitím speciálních znaků (mezer) ID “ 998” (“mezera-mezera-998)

|  |  |
| --- | --- |
| Kroky: | Očekávaný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek |
| 3. vybrat z nabídky DELETE (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)/ 998”,   * kliknout na tlačítko SEND | 3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND |
| 4. v části BODY se zobrazí vpravo error status code | 4. zobrazí se error status code 400 - špatný formát položky ID |
| 5. v části BODY v dolní části jsou doplňující informace k error statusu | 5. v části BODY v dolní části jsou doplňující informace k error statusu - “Špatný formát položka ID, položka ID nesmí obsahovat mezery.” |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id= 998  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Vložit uvedené příkaz a spustit jej prostřednictvím kombinace kláves CTRL+ enter. |
| 8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují a tudíž jej nelze smazat - zobrazí se error status, zadané ID má špatný formát | 8. v Workbench se zobrazí error status - špatný formát položky ID |

# EXEKUCE TESTŮ

|  |
| --- |
| **GET** |

1. GET ⇒ Abstrakt: test existujícího studenta **id: 219**

**Hodnoty v DTB:**  id: 219

age: 34

email: [jjohn34@yourmail.com](mailto:jjohn34@yourmail.com)

firstName: “ “

lastName: “JOSHUA”

|  |  |  |
| --- | --- | --- |
| *Kroky:* | *Očekávaný výsledek* | *Skutečný výsledek* |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 1. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a “HTTP” se v pořádku otevře nový požadavek | **OK** |
| 1. pod text “Untitled Request” vybrat z nabídky “GET” a do políčka napravo od něj vložit “<http://108.143.193.45:8080/api/v1/students/219>” a kliknu vpravo na tlačítko “SEND” | 3. po zvolení GET, vložení uvedeného URL a kliknutí na SEND se zobrazí v těle příslušné data pro studenta id 219 | **OK** |
| 1. Dole v sekci “BODY” na pravé straně se zobrazí status code “200 OK” |  | **OK** |
| 1. Pod sekci “BODY” se zobrazí příslušné údaje k id 219 ve formátu **JSON**:   {  "id": 219,  "firstName": "",  "lastName": "JOSHUA",  "email": "jjohn34@yourmail.com",  "age": 34  } |  | **BUG:**  *Hodnoty se zobrazí správně, ale chybí kontrola prázdných polí při zakládání viz. firstName.*  *Ověřit u zadavatele akceptaci prázdných hodnot v polích.*  - celý Printscreen pod touto tabulkou (č. printscreen GET č.1) |
| 1. otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB | **OK** |
| 1. Do části Query doplnit:   SELECT  \*  FROM student  WHERE id=219  ;  a spustit požadavek prostřednictvím CTRL+enter | 7. Zobrazí se příslušné údaje ke studentovi id 219:    id: 219  age: 34  email: “[jjohn34@yourmail.com](mailto:jjohn34@yourmail.com)”  first\_name: “ “  last\_name: “JOSHUA” | **BUG:**  *viz. bod 5 - ověřit u zadavatele akceptaci prázdných hodnot při zakládání*  - celý Printscreen pod touto tabulkou (č. printscreen GET č.1) |
| 1. Porovnat údaje z Postamen a MySQL WorkBench - zda jsou všechny shodné a zobrazily se správně:  * DTB id: **219** = Postmane id: **219** * DTB age: **34** = Postmane age: **34** * DTB email: [jjohn34@yourmail.com](mailto:jjohn34@yourmail.com) = Postamen email: [jjohn34@yourmail.com](mailto:jjohn34@yourmail.com) * DTB first\_name: “ “ = Postmane firstName: “ “ * DTB last\_name: “**JOSHUA**” = Postamen lastName: “**JOSHUA**” | 8. Všechny údaje z DTB se shodují s údaji v Postamen a zobrazují se kompletně:   * DTB id: **219** = Postmane id: **219** * DTB age: **34** = Postmane age: **34** * DTB email: [jjohn34@yourmail.com](mailto:jjohn34@yourmail.com) = Postamen email: [jjohn34@yourmail.com](mailto:jjohn34@yourmail.com) * DTB first\_name: “ “ = Postmane firstName: “ “ * DTB last\_name: “**JOSHUA**” = Postamen lastName: “**JOSHUA**” | **OK** |

|  |  |
| --- | --- |
| Printscreen GET abstrakt č. 1: | |
| Postman: | My SQL Workbench: |
|  |  |

2. GET ⇒ Abstrakt: test existujícího studenta **id: 307**

**Hodnoty v DTB:**  id: 307

age: 89

email: ýá+ěčzč51

firstName: “-86 “

lastName: “\*+6”

*POZN.: kombinace znaků a čísel v primárně textových polích, špatný formát emailu*

|  |  |  |
| --- | --- | --- |
| *Kroky:* | *Očekávaný výsledek:* | *Skutečný výsledek:* |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se bez problému otevře nový požadavek | **OK** |
| 3. pod text “Untitled Request” vybrat z nabídky “GET” a do políčka napravo zkopírovat tento odakaz “http://108.143.193.45:8080/api/v1/students/307” a kliknout vpravo na tlačítko “SEND” | 3. po zvolení GET, vložení uvedeného URL a kliknutí na SEND se zobrazí dole v části “BODY” příslušné data pro studenta id 307 | **OK** |
| 4. Dole v sekci “BODY” na pravé straně se zobrazí status code “200 OK” |  | **OK** |
| 5. Pod sekci “BODY” se zobrazí příslušné údaje k id 307 ve formátu **JSON**:  {  "id": 307,  "firstName": "-86",  "lastName": "\*+6",  "email": "ýá+ěčzč51",  "age": 89  } | - viz.Printscreen pod touto tabulkou (č. printscreen GET č. 2) | **BUG:**  Hodnoty se zobrazí správně, ale špatný formát atributů”:   * “email”, * “firstName” * “lastName”   - ověřit u zadavatele akceptaci čísel a znaků v polích firstName a lastName + ověřit kontrolu formátu e-mailové hodnoty |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem | **OK** |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=307  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Po spuštění příkazu se dole zobrazí v “Result Grid” příslušné údaje ke studentovi id 307:    id: 307  age: 89  email: “ýá+ěčzč51”  first\_name: “-86 “  last\_name: “\*+6” | **BUG:**  viz. bod. č. 5. - chybný formát atributů - “email”, “firstName” a “lastName” - čísla a symboly |
| 1. Porovnat údaje ke studentovi id 307 z Postman a MySQL WorkBench - zda jsou všechny shodné a zobrazily se správně:  * DTB id: **307** * DTB age: **89** * DTB email: ýá+ěčzč51 * DTB first\_name: “**-86**” * DTB last\_name: “**\*+6**” | 8. Všechny údaje z DTB se shodují s údaji v Postamen a zobrazují se kompletně:   * DTB id: **307 =** Postaman id: **307** * DTB age: **89** = Postman age: **89** * DTB email: ýá+ěčzč51 = Postman email: "ýá+ěčzč51" * DTB first\_name: “**-86**” = firstName: **"-86"** * DTB last\_name: “**\*+6**” = lastName: **"\*+6"** | **OK** |

|  |  |
| --- | --- |
| Printscreen GET č. 2: | |
| Postman: | My SQL Worbench: |
|  |  |

3. GET ⇒ Abstrakt: test neexistujícího studenta **id: 10**

*POZN.: student s tímto id neexistuje*

|  |  |  |
| --- | --- | --- |
| *Kroky:* | *Očekávaný výsledek:* | *Skutečný výsledek:* |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 1. po kliknutí na “NEW” a zvolení “HTTP” se bez problému otevře nový požadavek | **OK** |
| 3. pod text “Untitled Request” vybrat z nabídky “GET” a do políčka napravo vložit tento odkaz - “http://108.143.193.45:8080/api/v1/students/10” a kliknu vpravo na tlačítko “SEND” | 1. lze zvolit GET, vložit uvedeného URL a kliknout na SEND | **OK** |
| 1. V pravo v sekci “BODY” se objeví chybová hláška - chybový status code | 4. zobrazí se status code 404 - neexistující student | BUG:  zobrazí se status code 500 Internet Server Error (má se zobrazit status code 404) |
| 1. V sekci BODY se zobrazí potvrzení chybové hlášky | 5. v sekci BODY se zobrazí potvrzení chybové hlášky s upřesněním, že student s tímto ID neexistuje | BUG:  má se zobrazit status code 404 s doplňující informací - “Student s tímto ID neexistuje v databázi.” |
| 1. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem | ***OK*** |
| 1. Do části Query doplnit:   SELECT  \*  FROM student  WHERE id=10  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Po spuštění příkazu se dole v části “Result Grid” nezobrazí žádné informace, že student s ID 10 neexistuje a zároveň v části Action Output se zobrazí error “hláška” - student s tímto ID nenalezen. | **BUG:**   * správně se nezobrazí žádné informace o studentovi (protože student s tímto ID neexistuje),      * ale v části Action Output se nezobrazí error hláška, že student s tímto ID neexistuje - místo toho se zobrazí zobrazí “zelené fajfka”, že proběhl request v pořádku s 0 vrácenými řádky, což je v pořádku, nicméně měl by se zobrazit “červený křížek”, že student s tímto ID neexistuje |

|  |  |
| --- | --- |
| PrintScreen GET č. 3 | |
| Postman: | My SQL Workbench: |
|  |  |

4. GET ⇒ Abstrakt: test neexistujícího studenta id: **xy14**

*POZN.: neexistující ID studenta, ID v kombinaci znaků a čísel*

|  |  |  |
| --- | --- | --- |
| *Kroky:* | *Očekávaný výsledek:* | *Skutečný výsledek:* |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 1. po kliknutí na “NEW” a zvolení “HTTP” se bez problému otevře nový požadavek | **OK** |
| 3. pod text “Untitled Request” vyberu z nabídky “GET” a do políčka napravo od něj vložím “http://108.143.193.45:8080/api/v1/students/xy14” a kliknu vpravo na tlačítko “SEND” | 1. po zvolení GET, vložení uvedeného URL a kliknutí na SEND | **OK** |
| 4. V pravo v sekci “BODY” se objeví - chybový status code - 404 - neexistující student | 1. Zobrazí se chybová hláška - status code 404 - neexistující student | **BUG:**  zobrazí se status code 400 místo 404 neexistující student |
| 5. V sekci BODY se zobrazí potvrzení chybové hlášky | 1. zobrazí se status code 404 s doplňující informací - “Student s tímto ID neexistuje” | **BUG:**  má se zobrazit status code 404 s doplňující informací - “Student s tímto ID neexistuje”.  Ve skutečnosti se zobrazí viz. níže: |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem | **OK** |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=xy14  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Po spuštění příkazu se dole zobrazí v “Result Grid” nezobrazí žádné informace, protože student s ID xy14 neexistuje a zároveň v části Action Output se zobrazí error “hláška” | **OK** |

|  |  |
| --- | --- |
| PrintScreen GET č. 4: | |
| Postman: | MySQL Workbench: |
|  |  |

# 

|  |
| --- |
| **POST:** |

(požadavky - vše ukládat malými písmeny + povinné vyplněné všechny hodnoty v DTB)

1. POST ⇒ Abstrakt: založit studenta se všemi údaji “správně”- vyplněné všechny atributy ve správném formátu

**Hodnoty:**  id:

age: 39

email: pavel.nevimkdo@gmail.com

firstName: “ Pavel“

lastName: “Nevimkdo”

|  |  |  |
| --- | --- | --- |
| Kroky: | Očekávaný výsledek: |  |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek | **OK** |
| 3. vybrat z nabídky POST (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)”, | 3. lze vybrat variantu Post a vložit uvedené URL | **OK** |
| 4. Dole v části BODY, zakliknout možnost “raw”, rozkliknout možnost “TEXT” a zvolit variantu “JSON” | 4. v části BODY lze zakliknout možnost “raw” a vybrat variantu “JSON” v části “TEXT” | **OK** |
| 5. v nově otevřeném okně po zakliknutí “raw” doplnit tento text:  {  "firstName": "Pavel",  "lastName": "Nevimkdo",  "email": "pavel.nevimkdo@gmail.com",  "age": 39  }  a kliknout na SEND | 5. v pořádku lze vložit uvedené údaje a kliknout na “SEND” | **OK** |
| 6. zobrazí se status code 200 | 6. zobrazí se status code 200 | **OK** |
| 7. V spodní části BODY se zobrazí doplněné údaje ve formátu JSON a přidělí se ID student | 7. v spodní části se zobrazí veškeré zadané údaje a navíc se doplní automaticky ID studenta | **BUG:**  *Změnila se velikost písma v atributu “lastName” ze zadaného “Nevimkdo” na “NEVIMKDO” (vše velkými písmeny)* |
| 8. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 8. Otevřít aplikaci Workbench a připojit se na uvedenou DTB | **OK** |
| 9. do části Query doplnit tento text (s nově přiděleným ID k novému studentovi):  SELECT  \*  FROM student  WHERE id=972  ;  potvrdím CTRL+ENTER | 9. lze doplnit uvedené údaje a potvrdit CTRL+ENTER - zobrazí se údaje k studentovi | **OK** |
| 10. Porovnám údaje ke studentovi id 972 z Postamen a MySQL WorkBench - vše se shoduje, všechny atributy jsou zapsány správně - **malými písmeny a jsou vyplněné všechny atributy**: | 10. Všechny atributy k vytvořenému studentovi v DTB se shodují s údaji v Postman a zobrazují se kompletně, malými písmeny:   * DTB id: **972=** Postaman id: **972** * DTB age: **39** = Postman age: **39** * DTB email: **pavel.nevimkdo@gmail.com** = Postman email: **pavel.nevimkdo@gmail.com** * DTB first\_name: “**pavel**” = Postman firstName: **"Pavel"** * DTB last\_name: “**\*nevimkdo**” = lastName: **"\*Nevimkdo"** | **BUG:**  *Hodnoty v DTB se mají uložit malými písmeny - atributy “first\_name” (první písmeno) a “last\_name (všechny písmena - viz. bod 7) jsou však uložené velkými písmeny*. |

|  |  |
| --- | --- |
| Print | |
| Postman: | MySQL Workbench: |
|  |  |

1. POST ⇒ Abstrakt: založit studenta s chybějícími údaji ve jméně:

**Hodnoty:**  id:

age: 21

email: jsem.zde.kamo@seznam.cz

firstName: “ “

lastName: “Kamos”

|  |  |  |
| --- | --- | --- |
| Kroky: | Očekávaný výsledek: | Skutečný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek | **OK** |
| 3. vybrat z nabídky POST (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)”, | 3. lze vybrat variantu POST a vložit uvedené URL | **OK** |
| 4. Dole v části BODY, zakliknout možnost “raw”, rozkliknout možnost “TEXT” a zvolit variantu “JSON” | 4. v části BODY lze zakliknout možnost “raw” a vybrat variantu “JSON” v části “TEXT” | **OK** |
| 5. v nově otevřeném okně po zakliknutí “raw” doplnit tento text:  {  "firstName": " ",  "lastName": "Kamos",  "email": "jsem.zde.kamo@seznam.cz",  "age": 21  }  a kliknu na SEND | 5. v pořádku se zapíšou uvedené uvedené údaje a kliknout na “SEND” | **OK** |
| 6. zobrazí se status code 400 - který upozorní, že nejsou vyplněné všechny atrbuty | 6. zobrazí se error status code 400 s informací, že nejsou vyplněné všechny údaje. | **BUG:**  zobrazí se status code 200, že je vše v pořáku. Ve skutečnosti je prázdný atribut “firstName” a takovýto student by se neměl uložit. |
| 7. V spodní části BODY se zobrazí doplnění chybové hlášky, že není vyplněn atribut “firstName” | 7. v spodní části se zobrazí informace, že není vyplněn atribut “firstName” a je nutné jej doplnit - “Vyplňte položku jméno.” | **BUG:**  student s prázdným atributem “firstName” byl uložen a dokonce mu bylo přiděleno ID |
| 8. Postman studenta neuloží do DTB , dokud nebudou vyplněné všechny atributy. | 8. Postman tohoto studenta neuloží do DTB, dokud nebudou vyplněné všechny atributy včetně momentálně prázdného pole “firstName”. Studentovi s prázdným atributem “firstName” není přiděleno ID. | **BUG:**  Postman studenta uložil s přiděleným novým ID. Takovýto student byl následně uložen do DTB v WorkBench: |

3. POST ⇒ Abstrakt: založit studenta se špatným formátem v e-mailu a špatným formátem věku:

**Hodnoty:**  id:

age: -5

email: ahojkamocentrum.cz.com

firstName: “Jiřík “

lastName: “Kocourek”

|  |  |  |
| --- | --- | --- |
| Kroky: | Očekávaný výsledek: | Skutečný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek | **OK** |
| 3. vybrat z nabídky POST (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)”, | 3. lze vybrat variantu POST a vložit uvedené URL | **OK** |
| 4. Dole v části BODY, zakliknout možnost “raw”, rozkliknout možnost “TEXT” a zvolit variantu “JSON” | 4. v části BODY lze zakliknout možnost “raw” a vybrat variantu “JSON” v části “TEXT” | **OK** |
| 5. v nově otevřeném okně po zakliknutí “raw” doplnit tento text:  {  "firstName": "Jiřík ",  "lastName": "Kocourek",  "email": "ahojkamocentrum.cz.com",  "age": -5  }  a kliknu na SEND | 5. v pořádku lze vložit uvedené údaje a kliknout na “SEND” | **OK** |
| 6. zobrazí se status code 400 - který upozorní, že atribut “age” a “email” mají špatný formát. | 6. zobrazí se error status code 400 s informací, že atribut “age” a “email” jsou ve špatném formátu. | **BUG:**  zobrazí se status code 200 |
| 7. V spodní části BODY se zobrazí doplnění chybové hlášky, že atributy “age” a “email” mají špatný formát a je nutné je vyplnit správně. | 7. v spodní části se zobrazí informace, že není atribut “age” a “email” jsou ve špatném formátu. Je nutné je vyplnit správně - “Vyplňte hodnoty věk a e-mail ve správném formátu” | **BUG:**  student s nesprávnými formáty atributů “age” a “email” byl uložen a dokonce mu bylo přiděleno ID |
| 8. Postman studenta neuloží, dokud nebudou vyplněné všechny atributy ve správném formátu. | 8. Postman tohoto studenta neuloží, dokud nebudou vyplněné správně všechny atributy včetně momentálně špatně zadaného atributu “age” = -5(záporná hodnota) a atributu “email” - chybí @. | **BUG:**  Postman studenta uložil s přiděleným novým ID (přestože je špatný formát atributů “age” a “email”). Takovýto student byl následně uložen do DTB v WorkBench: |

4. POST ⇒ Abstrakt: založit totožného studenta jako v POST abstrakt 1 - všechny hodnoty totožné (test duplicit)

(Postman by měl upozornit, že student s těmito údaji je již založen - kontrola duplicitního uložení studentu dle dokumentace - pro tyto potřeby kontrola dle e-mailu)

**Hodnoty:**  id:

age: 39

email: pavel.nevimkdo@gmail.com

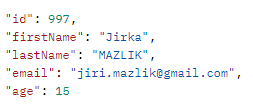
firstName: “ Pavel“

lastName: “Nevimkdo”

|  |  |  |
| --- | --- | --- |
| Kroky: | Očekávaný výsledek: | Skutečný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek | **OK** |
| 3. vybrat z nabídky POST (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)”, | 3. lze vybrat variantu POST a vložit uvedené URL | **OK** |
| 4. Dole v části BODY, zakliknout možnost “raw”, rozkliknout možnost “TEXT” a zvolit variantu “JSON” | 4. v části BODY lze zakliknout možnost “raw” a vybrat variantu “JSON” v části “TEXT” | **OK** |
| 5. v nově otevřeném okně po zakliknutí “raw” doplnit tento text:  {  "firstName": "Pavel",  "lastName": "Nevimkdo",  "email": "pavel.nevimkdo@gmail.com",  "age": 39  }  a kliknu na SEND | 5. v pořádku se zapíšou uvedené uvedené údaje a kliknout na “SEND” | **OK** |
| 6. zobrazí se status code 400 - který upozorní, že student s tímto email již existuje | 6. zobrazí se error status code 400 s informací, že student s tímto s tímto emailem existuje | **BUG:**  zobrazí se status code 200 |
| 7. V spodní části BODY se zobrazí doplnění chybové hlášky, že studenta nelze uložit, protože student s tímto e-mailem již existuje. | 7. v spodní části se zobrazí informace, že student s těmito údaji je již založen. | **BUG:**  Postman studenta uloží, přestože je již totožný student (totožný email) v DTB založen. Studentovi je přiděleno i ID: |
| 8. Postman studenta neuloží, hlídá duplicitní uložení studentů dle jedinečné hodnoty v atributu “email” - dle dokumentace. | 8. Postman studenta neuloží, hlídá duplicitní uložení studentů dle jedinečné hodnoty v atributu “email” - dle dokumentace. Pokud je student pod tímto e-mailem již založen, nelze pod totožným emailem zakládat dalšího studenta. Kontrola duplicit dle dokumentace (e-mail, příjmení, jméno….) | **BUG:**  Postman studenta uložil s přiděleným novým ID (přestože již tento student s tímto email je v DTB uložen). Takovýto student byl následně chybně uložen do DTB v WorkBench:    Takže máme uloženého jednoho studenta v DTB 2x pod různým ID: |

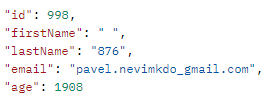
|  |
| --- |
| **DELETE:** |

1. DELETE ⇒ Abstrakt: smazání studenta ID 997 (všechny atributy vyplněné)

****

|  |  |  |
| --- | --- | --- |
| Kroky: | Očekávaný výsledek: | Skutečný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek | **OK** |
| 3. vybrat z nabídky DELETE (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)/997”,   * kliknout na tlačítko SEND | 3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND | **OK** |
| 4. v části BODY se zobrazí vpravo status code 200 | 4. zobrazí se status code 200 | **OK** |
| 5. v části BODY v dolní části nejsou vyplněné žádné údaj - student je smazán | 5. v části BODY nejsou žádné údaje - student je smazán | **OK** |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem | **OK** |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=997  ;  a spustit požadavek prostřednictvím CTRL+enter | 7. Vložit uvedené příkaz a spustit jej prostřednictvím kombinace kláves CTRL+ enter. | **OK** |
| 8. v Workbench se nezobrazí žádné údaje, protože student je smazán | 8. v Workbench se nezobrazí žádné údaje, protože student je smazán | **OK** |

1. DELETE ⇒ Abstrakt: smazání existujícího studenta ID 998 (chybí vyplněný atribut “firstName”)





|  |  |  |
| --- | --- | --- |
| Kroky: | Očekávaný výsledek: | Skutečný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek | **OK** |
| 3. vybrat z nabídky DELETE (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)/998”,   * kliknout na tlačítko SEND | 3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND | **OK** |
| 4. v části BODY se zobrazí vpravo status code 200 | 4. zobrazí se status code 200 | **OK** |
| 5. v části BODY v dolní části nejsou vyplněné žádné údaj - studnt je smazán | 5. v části BODY nejsou žádné údaje - student je smazán | **OK** |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem | **OK** |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=998  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Vložit uvedené příkaz a spustím jej prostřednictvím kombinace kláves CTRL+ enter. | **OK** |
| 8. v Workbench se nezobrazí žádné údaje, protože student je smazán | 8. v Workbench se nezobrazí žádné údaje, protože student je smazán | **OK** |

3. DELETE ⇒ Abstrakt: smazání studenta s neexistujícím ID - 1234

|  |  |  |
| --- | --- | --- |
| Kroky: | Očekávaný výsledek: | Očekávaný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek | **OK** |
| 3. vybrat z nabídky DELETE (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)/1234”,   * kliknout na tlačítko SEND | 3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND | **OK** |
| 4. v části BODY se zobrazí vpravo error status code | 4. zobrazí se error status code 404 - neexistující student resp. ID | **BUG:**  má se zobrazit status code 404 - neexistující student resp. ID.  Místo toho se zobrazil status code 500 |
| 5. v části BODY v dolní části jsou doplňující informace k error statusu | 5. v části BODY v dolní části jsou doplňující informace k error statusu - “Student s tímto ID není uložen v databázi”, | **BUG:**  neuvedla se chybová hláška “Student s tímto ID není uložen v databázi”.  Ve skutečnosti se zobrazila tato informace - viz. níže printscreen: |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem | **OK** |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=1234  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Vložit uvedené příkaz a spustit jej prostřednictvím kombinace kláves CTRL+ enter. | **OK** |
| 8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují a tudíž jej nelze smazat | 8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují a tudíž jej nelze smazat | **OK** |

4. DELETE ⇒ Abstrakt: smazání studenta s neexistujícím ID s použitím speciálních znaků (podtržítko) ID 99\_8

|  |  |  |
| --- | --- | --- |
| Kroky: | Očekávaný výsledek: | Skutečný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek | **OK** |
| 3. vybrat z nabídky DELETE (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)/99\_8”,   * kliknout na tlačítko SEND | 3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND | **OK** |
| 4. v části BODY se zobrazí vpravo error status code | 4. zobrazí se error status code 400 - špatný format atributu ID | **OK** |
| 5. v části BODY v dolní části jsou doplňující informace k error statusu | 5. v části BODY v dolní části jsou doplňující informace k error statusu - “Špatný formát položky ID, zadejte pouze čísla” | BUG:   * nezobrazí se message “Špatný formát položky ID, zadejte pouze čísla”. |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem |  |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id=99\_8  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Vložit uvedené příkaz a spustit jej prostřednictvím kombinace kláves CTRL+ enter. | **OK** |
| 8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují, ID má špatný formát a tudíž jej nelze smazat - zobrazí se error status | 8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují a tudíž jej nelze smazat - zobrazí se error status | **OK** |

# 

5. DELETE ⇒ Abstrakt: smazání studenta s neexistujícím ID s použitím speciálních znaků (mezer) ID “ 998” (“mezera-mezera-998)

|  |  |  |
| --- | --- | --- |
| Kroky: | Očekávaný výsledek: | Skutečný výsledek: |
| 1. otevřít aplikaci Postman | 1. aplikace Postman se v pořádku otevře | **OK** |
| 2. kliknout vlevo nahoře na “NEW”, zvolit “HTTP” a otevře se nový požadavek | 2. po kliknutí na “NEW” a zvolení “HTTP” se otevře nový požadavek | **OK** |
| 3. vybrat z nabídky DELETE (pod části Untitled Request) a vložit do kolonky vedle tento URL: “[http://108.143.193.45:8080/api/v1/students](http://108.143.193.45:8080/api/v1/students/xy)/ 998”,   * kliknout na tlačítko SEND | 3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND | **OK** |
| 4. v části BODY se zobrazí vpravo error status code | 4. zobrazí se error status code 400 - špatný formát položky ID | BUG:   * zobrazí se status code 500 namísto codu 400 špatný požadavek, ID obsahuje nesprávné znaky |
| 5. v části BODY v dolní části jsou doplňující informace k error statusu | 5. v části BODY v dolní části jsou doplňující informace k error statusu - “Špatný formát položky ID, položka ID nesmí obsahovat mezery.” | BUG:   * chybí message - “Špatný formát položky ID, položka nesmí obsahovat mezery”. |
| 6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB “engeto\_databse\_testing” | 6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem | **OK** |
| 7. Do části Query doplnit:  SELECT  \*  FROM student  WHERE id= 998  ;  a spustím požadavek prostřednictvím CTRL+enter | 7. Vložit uvedené příkaz a spustit jej prostřednictvím kombinace kláves CTRL+ enter. | **OK** |
| 8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují a tudíž jej nelze smazat - zobrazí se error status, zadané ID má špatný formát | 8. v Workbench se zobrazí error status - špatný formát položky ID | BUG:   * požadavek proběhne, neobjeví se error status * WorkBench nevidí mezery v příkazu a vezme pouze čísla, důležité zajistit při zakládání nových dat v Postmanu kontrolu, že je ID ve správném formátu bez mezer, znaků atd. |

# 

# BUG REPORT

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

|  |
| --- |
| **GET** |

* GET ⇒ abstrakt č. 1 - viz. bod 5 a 7 exekuce testu

Hodnoty se zobrazí správně, ale chybí kontrola prázdných polí při zakládání nového studenta viz. firstName. Ověřit u zadavatele akceptaci prázdných hodnot v jednotlivých polích při zakládání studenta.

* GET ⇒ abstrakt č. 2 - viz. bod 5 a 7 exekuce testu

Hodnoty se zobrazí správně, ale špatný formát atributů:

* **email** (chybný formát - chybí @ a . )
* **firstName** (čísla a znaky v tomto atributu)
* **lastName** (čísla a znaky v tomto atributu)

Ověřit u zadavatele akceptaci čísel a znaků v polích “firstName” a “lastName” a ověřit kontrolu správného formátu e-mailu (chybí @ a “tečka”)

* GET ⇒ abstrakt č. 3 - viz. bod 4, 5 a 7 exekuce testu
  + bod 4 - u neexistujícího studenta se má zobrazit status code 404, zde se však zobrazí status code 500 - Internet Server Error
  + bod 5 - má se zobrazit status code 404 s doplňující informací, že “Student s uvedeným ID neexistuje v databázi”
  + bod 7 - správně se nezobrazí žádné informace o studentovi (protože student s tímto ID neexistuje),ale v části Action Output se nezobrazí error hláška, že student s tímto ID neexistuje - místo toho se zobrazí zobrazí “zelené fajfka”, že proběhl request v pořádku s 0 vrácenými řádky, což je v pořádku, nicméně měl by se zobrazit “červený křížek”, že student s tímto ID neexistuje
* GET ⇒ abstrakt č. 4 - viz. bod 4 a 5 exekuce testu
  + bod 4 -zobrazí se status code 400 místo 404 - neexistující student
  + bod 5 -má se zobrazit status code 404 s doplňující informací - “Student s tímto ID neexistuje”.

|  |
| --- |
| **POST:** |

* POST ⇒ abstrakt č. 1 - viz. bod 7 a 10 exekuce testu
  + bod 7 - Postman - změnila se velikost písma v atributu “lastName” ze zadaného “Nevimkdo” na “NEVIMKDO” (vše velkými písmeny)
  + bod 10 - My SQL Workbench - hodnoty v DTB se mají uložit malými písmeny - atributy “first\_name” a “last\_name\_ jsou však uložené velkými písmeny.
* POST ⇒ abstrakt č. 2 - viz. body 6, 7 a 8 exekuce testu
  + bod 6 - Postman - zobrazí se status code 200, že je vše v pořádku. Ve skutečnosti je prázdný atribut “firstName” a takovýto student by se neměl uložit.
  + bod 7 - Postman - student s prázdným atributem “firstName” byl uložen a dokonce mu bylo přiděleno ID
  + bod 8 - Postman studenta uložil s přiděleným novým ID. Takovýto student byl následně uložen do DTB v WorkBench
* POST ⇒ abstrakt č. 3 - viz. body 6, 7 a 8 exekuce testu
  + bod 6 - má se zobrazit status code 400 s informací, že je chybný formát e-mail a age, ale ve skutečnosti se zobrazil status code 200
  + bod 7 - student s nesprávnými formáty atributů “age” a “email” byl uložen a dokonce mu bylo přiděleno ID
  + bod 8 - Postman studenta uložil s přiděleným novým ID (přestože je špatný formát atributů “age” a “email”). Takovýto student byl následně uložen do DTB v WorkBench:
* POST ⇒ abstrakt č. 4 - viz. body 6, 7, a 8 exekuce testu
  + bod 6 - zobrazí se status code 200, má se zobrazit status 400 (kontrola duplicit)
  + bod 7 - Postman studenta uloží, přestože je již totožný student (totožný email) v DTB založen. Studentovi je přiděleno i nové ID
  + bod 8 - Postman studenta uložil s přiděleným novým ID (přestože již tento student s tímto email je v DTB uložen). Takovýto student byl následně chybně uložen do DTB v WorkBench. Takže je jeden student uložen 2x pod různým ID.

|  |
| --- |
| **DELETE:** |

* DELETE ⇒ abstrakt č. 3 - viz. body 4, 5 exekuce testu
  + bod 4 - měl se zobrazit status code 404 - neexistující student resp. ID. Zobrazí se chybně status code 500.
  + bod 5 - neuvedla se chybová hláška “Student s tímto ID není uložen v databázi”.
* DELETE ⇒ abstrakt č. 4 - viz. bod 5 exekuce testu
* bod 4 - nezobrazí se message “Špatný formát položky ID, zadejte pouze čísla”.
* DELETE ⇒ abstrakt č. 5 - viz. bod 4, 5 a 8 exekuce testu
* bod 4 - zobrazí se status code 500 namísto codu 400 špatný požadavek, ID obsahuje nesprávné znaky
* bod 5 - chybí message - “Špatný formát položky ID, položka nesmí obsahovat mezery”.
* bod 8 - požadavek proběhne, neobjeví se error status. WorkBench nevidí mezery v příkazu a vezme pouze čísla (tedy vidí ID=998, namísto ID=” 998”), důležité zajistit při zadávání v Postmanu kontrolu, že je ID ve správném formátu bez mezer, znaků atd.