

FINÁLNÍ PROJEKT

č.1



ENGETO

Autor: Gabriela Řempělová
Datum:

OBSAH

ZADÁNÍ	3
TESTOVACÍ SCÉNÁŘE	4
EXEKUCE TESTŮ	5
BUG REPORT	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: 68q0llcbuyijdt5mzq2 Password: pscale_pw_zxfenAXfSYx9loe9PCS1snuGKr Fzw2J84BMIVZI96o
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



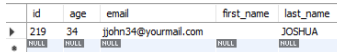
GET:

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

Hodnoty v DTB: id: 219
age: 34
email: jjohn34@yourmail.com
firstName: " "
lastName: "JOSHUA"

Pozn. prázdná hodnota v poli 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 bez problému otevře nový požadavek
3. 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

<p>4. Dole v sekci "BODY" na pravé straně se zobrazí status code "200 OK"</p>	
<p>5. Pod sekci "BODY" se zobrazí příslušné údaje k id 219 ve formátu JSON:</p> <pre>{ "id": 219, "firstName": "", "lastName": "JOSHUA", "email": "jjohn34@yourmail.com", "age": 34 }</pre>	
<p>6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB "engeto_databse_testing"</p>	<p>6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem</p>
<p>7. Do části Query doplnit:</p> <pre>SELECT * FROM student WHERE id=219 ;</pre> <p>a spustím požadavek prostřednictvím CTRL+enter</p>	<p>7. Po spuštění příkazu se dole zobrazí v "Result Grid" příslušné údaje ke studentovi id 219:</p>  <pre>id: 219 age: 34 email: "jjohn34@yourmail.com" first_name: "" last_name: "JOSHUA"</pre>
<p>8. Porovnat údaje ke studentovi id 219 z Postamen a MySQL WorkBench - zda jsou všechny shodné a zobrazily se správně:</p> <ul style="list-style-type: none"> DTB id: 219 = Postmane id: 219 	<p>Všechny údaje z DTB se shodují s údaji v Postamen a zobrazují se kompletně:</p> <ul style="list-style-type: none"> DTB id: 219 = Postmane id: 219

<ul style="list-style-type: none"> • DTB age: 34 = Postmane age: 34 • DTB email: jjohn34@yourmail.com = Postamen email: jjohn34@yourmail.com • DTB first_name: " " = Postmane firstName: " " • DTB last_name: "JOSHUA" = Postamen lastName: "JOSHUA" 	<ul style="list-style-type: none"> • DTB age: 34 = Postmane age: 34 • DTB email: jjohn34@yourmail.com = Postamen email: jjohn34@yourmail.com • DTB first_name: " " = Postmane firstName: " " • DTB last_name: "JOSHUA" = Postamen lastName: "JOSHUA"
--	--

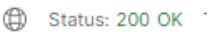
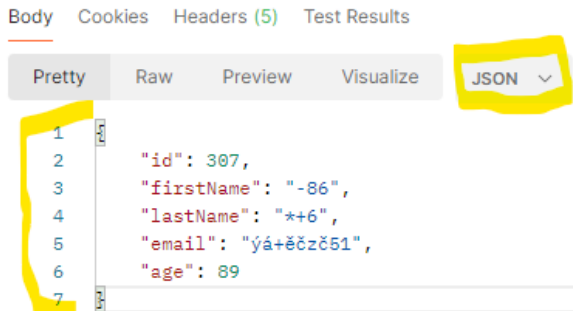
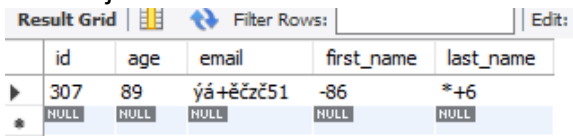
2. GET ⇒ Abstrakt: test existujícího studenta [id: 307](#)

Hodnoty v DTB:

- id: 307
- age: 89
- email: ýá+ěčzč51
- firstName: "-86 "
- lastName: "*+6"

POZN.: kombinace znaků a čísel v primárně textových polích

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 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/307 " a kliknout	3. po zvolení GET, vložení uvedeného URL a kliknutí na SEND se zobrazí dole v části "BODY" příslušné data pro studenta id 307

vpravo na tlačítko "SEND"	
4. Dole v sekci "BODY" na pravé straně se zobrazí status code "200 OK"	
5. Pod sekci "BODY" se zobrazí příslušné údaje k id 307 ve formátu JSON : <pre> { "id": 307, "firstName": "-86", "lastName": "*+6", "email": "ýá+ěčzč51", "age": 89 } </pre>	
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: <pre> SELECT * FROM student WHERE id=307 ; </pre> a spustit požadavek prostřednictvím CTRL+enter	7. Po spuštění příkazu se dole zobrazí v "Result Grid" příslušné údaje ke studentovi id 307:  <pre> id: 307 age: 89 email: "ýá+ěčzč51" first_name: "-86 " last_name: "*+6" </pre>

<p>8. Porovnat údaje ke studentovi id 307 z Postamen a MySQL WorkBench - zda jsou všechny shodné a zobrazily se správně:</p> <ul style="list-style-type: none"> • DTB id: 307 • DTB age: 89 • DTB email: ýá+ěččč51 • DTB first_name: "-86" • DTB last_name: "*+6" 	<p>8. Všechny údaje z DTB se shodují s údaji v Postamen a zobrazují se kompletně:</p> <ul style="list-style-type: none"> • DTB id: 307 = Postman id: 307 • DTB age: 89 = Postman age: 89 • DTB email: ýá+ěččč51 = Postman email: "ýá+ěččč51" • DTB first_name: "-86" = firstName: "-86" • DTB last_name: "*+6" = lastName: "*+6"
--	---

3. 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	2. 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"	3. lze vybrat GET, vložit uvedeného URL a kliknout na SEND
4. V pravo v sekci "BODY" se objeví chybová hláška	4. zobrazí se chybová hláška 404 - neexistující student resp. student

	s tímto ID neexistuje
5. 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
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=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.

4. 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	2. 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 -	3. lze zvolit GET, vložit uvedené URL a kliknout na tlačítko SEND

<p>"http://108.143.193.45:8080/api/v1/students/xy14" a kliknu vpravo na tlačítko "SEND"</p>	
<p>4. V dolní části se objeví vpravo chybová hláška - chybový status code - 404 - neexistující student</p>	<p>4. Zobrazí se chybová hláška - status code 404 - neexistující student resp. student s tímto ID neexistuje</p>
<p>5. Dole v sekci "BODY" se objeví doplňující chybová hláška</p>	<p>5. v sekci BODY se zobrazí potvrzení chybové hlášky s upřesněním, že student s tímto ID neexistuje</p>
<p>6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB "engeto_databse_testing"</p>	<p>6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem</p>
<p>7. Do části Query doplnit:</p> <pre>SELECT * FROM student WHERE id=xy14 ;</pre> <p>a spustím požadavek prostřednictvím CTRL+enter</p>	<p>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"</p>

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
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 ",	3. lze vybrat variantu POST a vložit uvedené URL
4. 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"
5. v nově otevřeném okně po zakliknutí "raw" doplnit tento text: { "firstName": "Pavel", "lastName": "Nevimkdo",	5. v pořádku lze zapsat uvedené údaje a kliknout na "SEND"

<p>"email": "pavel.nevimkdo@gmail.com", "age": 39 } a kliknu na SEND</p>	
6. zobrazí se status code 200	6. zobrazí se status code 200
7. 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
8. 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
9. 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
10. 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 : <ul style="list-style-type: none"> • 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"

2. 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" ,	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	5. v pořádku se zapíší uvedené údaje a lze kliknout na "SEND"

} a kliknu na SEND	
6. zobrazí se status code 400 - který upozorní, že nejsou vyplněné všechny atributy	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.

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:
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" ,	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: <pre>{ "firstName": "Jiřík ", "lastName": "Kocourek", "email": "ahojkamocentrum.cz.com", "age": -5 }</pre> a kliknu na SEND	5. v pořádku se zapíšou 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 @.

4. 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 této URL: "http://108.143.193.45:8080/api/v1/students" ,	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",	5. v pořádku se zapíšou uvedené údaje a kliknout na "SEND"

<pre>"lastName": "Nevimkdo", "email": "pavel.nevimkdo@gmail.com", "age": 39 }</pre> <p>a kliknu na SEND</p>	
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)

```
"id": 997,  
"firstName": "Jirka",  
"lastName": "MAZLIK",  
"email": "jiri.mazlik@gmail.com",  
"age": 15
```

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/997" , - kliknout na tlačítko SEND	3. lze vybrat variantu DELETE a vložit uvedené URL a 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 údaje - 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

<p>7. Do části Query doplnit:</p> <pre>SELECT * FROM student WHERE id=997 ;</pre> <p>a spustím požadavek prostřednictvím CTRL+enter</p>	<p>7. Vložit uvedené příkaz a spustím prostřednictvím požadavek přes kombinace kláves CTRL+ enter.</p>
<p>8. v Workbench se nezobrazí žádné údaje, protože student je smazán</p>	<p>8. v Workbench se nezobrazí žádné údaje, protože student je smazán</p>

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

```
"id": 998,
"firstName": " ",
"lastName": "876",
"email": "pavel.nevimkdo_gmail.com",
"age": 1908
```

id	age	email	first_name	last_name
998	1908	pavel.nevimkdo_gmail.com		876
NULL	NULL	NULL	NULL	NULL

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/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 údaje - 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=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/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 error 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: <code>SELECT * FROM student WHERE id=1234 ;</code> 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 neexistuje a tudíž jej nelze smazat	8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistuje 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/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: <code>SELECT * FROM student WHERE id=99_8 ;</code>	7. Vložit uvedené příkaz a spustit jej prostřednictvím kombinace kláves CTRL+ enter.

a spustím požadavek prostřednictvím 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/ 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	6. aplikace se bez problému otevře s připojením na uvedenou

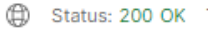
je uvedeno připojení na DTB "engeto_databse_testing"	DTB a se záložkou Query s prázdným tělem
<p>7. Do části Query doplnit:</p> <pre>SELECT * FROM student WHERE id= 998 ;</pre> <p>a spustím požadavek prostřednictvím CTRL+enter</p>	<p>7. Vložit uvedené příkaz a spustit jej prostřednictvím kombinace kláves CTRL+ enter.</p>
<p>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</p>	<p>8. v Workbench se zobrazí error status - špatný formát položky ID</p>

EXEKUCE TESTŮ

GET

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

Hodnoty v DTB: id: 219
 age: 34
 email: jjohn34@yourmail.com
 firstName: " "
 lastName: "JOSHUA"

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 "HTTP" se v pořádku otevře nový požadavek	OK
3. 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
4. Dole v sekci "BODY" na pravé straně se zobrazí status code "200 OK"		OK

<p>5. Pod sekci "BODY" se zobrazí příslušné údaje k id 219 ve formátu JSON:</p> <pre>{ "id": 219, "firstName": "", "lastName": "JOSHUA", "email": "jjohn34@yourmail.com", "age": 34 }</pre>	<p>body Cookies Headers (5) Test Results</p> <p>Pretty Raw Preview Visualize JSON</p> <pre>1 2 "id": 219, 3 "firstName": "", 4 "lastName": "JOSHUA", 5 "email": "jjohn34@yourmail.com", 6 "age": 34 7</pre>	<p>BUG:</p> <p><i>Hodnoty se zobrazí správně, ale chybí kontrola prázdných polí při zakládání viz. firstName.</i></p> <p><i>Ověřit u zadavatele akceptaci prázdných hodnot v polích.</i></p> <p>- celý Printscreen pod touto tabulkou (č. printscreen GET č.1)</p>										
<p>6. otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB "engeto_databse_testing"</p>	<p>6. aplikace se bez problému otevře s připojením na uvedenou DTB</p>	<p>OK</p>										
<p>7. Do části Query doplnit:</p> <pre>SELECT * FROM student WHERE id=219 ;</pre> <p>a spustit požadavek prostřednictvím CTRL+enter</p>	<p>7. Zobrazí se příslušné údaje ke studentovi id 219:</p> <table><thead><tr><th>id</th><th>age</th><th>email</th><th>first_name</th><th>last_name</th></tr></thead><tbody><tr><td>219</td><td>34</td><td>jjohn34@yourmail.com</td><td>JOSHUA</td><td></td></tr></tbody></table> <p>id: 219 age: 34 email: "jjohn34@yourmail.com" first_name: "" last_name: "JOSHUA"</p>	id	age	email	first_name	last_name	219	34	jjohn34@yourmail.com	JOSHUA		<p>BUG:</p> <p><i>viz. bod 5 - ověřit u zadavatele akceptaci prázdných hodnot při zakládání</i></p> <p>- celý Printscreen pod touto tabulkou (č. printscreen GET č.1)</p>
id	age	email	first_name	last_name								
219	34	jjohn34@yourmail.com	JOSHUA									
<p>8. Porovnat údaje z Postamen a MySQL WorkBench - zda jsou všechny shodné a zobrazily se správně:</p>	<p>8. Všechny údaje z DTB se shodují s údaji v Postamen a zobrazují se kompletně:</p>	<p>OK</p>										

- DTB id: **219** = Postmane id: **219**
- DTB age: **34** = Postmane age: **34**
- DTB email: jjohn34@yourmail.com =
Postamen email: jjohn34@yourmail.com
- DTB first_name: " " = Postmane firstName: " "
- DTB last_name: "**JOSHUA**" = Postamen
lastName: "**JOSHUA**"

- DTB id: **219** = Postmane id: **219**
- DTB age: **34** = Postmane age: **34**
- DTB email: jjohn34@yourmail.com =
Postamen email: jjohn34@yourmail.com
- DTB first_name: " " = Postmane firstName: " "
- DTB last_name: "**JOSHUA**" = Postamen
lastName: "**JOSHUA**"

Printscreen GET abstrakt č. 1:

Postman:

GET http://108.143.193.45:8080/api/v1/students/219

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```

1 {
2   "id": 219,
3   "firstName": "",
4   "lastName": "JOSHUA",
5   "email": "jjohn34@yourmail.com",
6   "age": 34
7 }
```

My SQL Workbench:

Limit to 1000 rows

```

1 SELECT
2   *
3 FROM student
4 WHERE id=219;
5
6
7
```

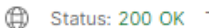
Result Grid

id	age	email	first_name	last_name
219	34	jjohn34@yourmail.com		JOSHUA
NULL	NULL	NULL	NULL	NULL

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

Hodnoty v DTB: id: 307
age: 89
email: ýá+ěčžč51
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 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	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",	- viz.Printscreen pod touto tabulkou (č. printscreen GET č. 2)	BUG: Hodnoty se zobrazí správně, ale špatný formát atributů: <ul style="list-style-type: none">• "email",• "firstName"• "lastName"

<pre>"lastName": "*+6", "email": "ýá+ěčzč51", "age": 89 }</pre>	<div>Body Cookies Headers (5) Test Results</div> <div>Pretty Raw Preview Visualize JSON</div> <div><pre>1 2 "id": 307, 3 "firstName": "-86", 4 "lastName": "*+6", 5 "email": "ýá+ěčzč51", 6 "age": 89 7</pre></div>	<p>- ověřit u zadavatele akceptaci čísel a znaků v polích firstName a lastName + ověřit kontrolu formátu e-mailové hodnoty</p>																		
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: <div>Result Grid Filter Rows: Edit:</div> <table><thead><tr><th></th><th>id</th><th>age</th><th>email</th><th>first_name</th><th>last_name</th></tr></thead><tbody><tr><td>▶</td><td>307</td><td>89</td><td>ýá+ěčzč51</td><td>-86</td><td>*+6</td></tr><tr><td>*</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></tbody></table> id: 307 age: 89 email: "ýá+ěčzč51" first_name: "-86 " last_name: "*+6"		id	age	email	first_name	last_name	▶	307	89	ýá+ěčzč51	-86	*+6	*	NULL	NULL	NULL	NULL	NULL	BUG: viz. bod. č. 5. - chybný formát atributů - "email", "firstName" a "lastName" - čísla a symboly
	id	age	email	first_name	last_name															
▶	307	89	ýá+ěčzč51	-86	*+6															
*	NULL	NULL	NULL	NULL	NULL															
9. 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	8. Všechny údaje z DTB se shodují s údaji v Postman a zobrazují se kompletně: • DTB id: 307 = Postman id: 307 • DTB age: 89 = Postman age: 89	OK																		

- DTB email: yá+ěčzč51
- DTB first_name: **"-86"**
- DTB last_name: **"*+6"**

- DTB email: ýá+ěčzc51 = Postman email: "ýá+ěčzc51"
- DTB first_name: **"-86"** = firstName: **"-86"**
- DTB last_name: **"*+6"** = lastName: **"*+6"**

Printscreen GET č. 2:

Postman:

GET

▼

http://108.143.193.45:8080/api/v1/students/307

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

Key	Value
Key	Value

Body

Cookies

Headers (5)

Test Results

Status: 200

Pretty

Raw

Preview

Visualize

JSON ▼

⋮

```

1
2  "id": 307,
3  "firstName": "-86",
4  "lastName": "*+6",
5  "email": "ýá+ëçzë51",
6  "age": 89
7

```

My SQL Worbench:

```
1  SELECT
2      *
3  FROM student
4  WHERE id=307;
```

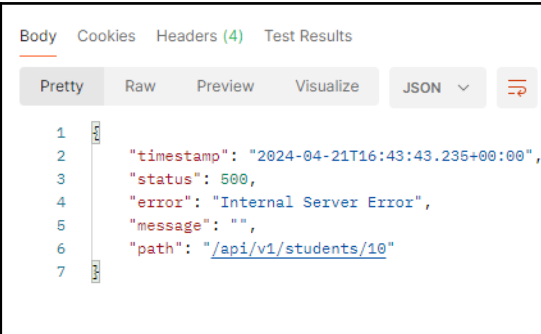
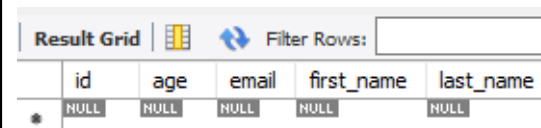
Result Grid

	id	age	email	first_name	last_name
▶	307	89	ýá+ěčzč51	-86	*+6
*	NULL	NULL	NULL	NULL	NULL

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	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 vložit tento odkaz - " http://108.143.193.45:8080/api/v1/students/10 " a kliknu vpravo na tlačítko "SEND"	3. lze zvolit GET, vložit uvedeného URL a kliknout na SEND	OK
4. 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) Status: 500 Internal Server Error
5. 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."

		 <pre> 1 { 2 "timestamp": "2024-04-21T16:43:43.235+00:00", 3 "status": 500, 4 "error": "Internal Server Error", 5 "message": "", 6 "path": "/api/v1/students/10" 7 } </pre>
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: <pre> SELECT * FROM student WHERE id=10 ; </pre> 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.	<p>BUG:</p> <ul style="list-style-type: none"> - správně se nezobrazí žádné informace o studentovi (protože student s tímto ID neexistuje),  <p>-</p> <ul style="list-style-type: none"> - ale v části Action Output se nezobrazí error hláška, že student s tímto ID neexistuje - místo toho se 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

		41 10:24:33 SELECT * FROM student WHERE id=10 LIMIT 0, 1000 0 row(s) returned
--	--	---

PrintScreen GET č. 3

Postman:

GET http://108.143.193.45:8080/api/v1/students/10

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

ody Cookies Headers (4) Test Results 500 Internal Server Error

Pretty Raw Preview Visualize JSON

```

1  {
2    "timestamp": "2024-04-25T16:22:23.334+00:00",
3    "status": 500,
4    "error": "Internal Server Error",
5    "message": "",
6    "path": "/api/v1/students/10"
7  }
```

My SQL Workbench:

1 select

2 *

3 from student

4 where id = 10;


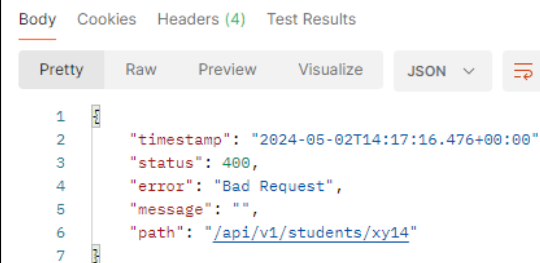
Result Grid

	id	age	email	first_name	last_name
*	NULL	NULL	NULL	NULL	NULL

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	2. 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"	3. 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	4. Zobrazí se chybová hláška - status code 404 - neexistující student	<p>BUG: zobrazí se status code 400 místo 404 neexistující student</p> 
5. V sekci BODY se zobrazí potvrzení chybové hlášky	5. zobrazí se status code 404 s doplňující informací - "Student s tímto ID neexistuje"	<p>BUG: má se zobrazit status code 404 s doplňující informací - "Student s tímto ID neexistuje".</p> <p>Ve skutečnosti se zobrazí viz. níže:</p> 

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: <pre>SELECT * FROM student WHERE id=xy14 ;</pre> 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 <div> <div>9 16:19:15 SELECT * FROM student WHERE id=xy14 LIMIT 0, 1000</div> <div>Error Code: 1054, target: qa_demo - primary: vtable: rpc error: code = NotFound desc = Unk...</div> </div>

PrintScreen GET č. 4:

Postman:	MySQL Workbench:

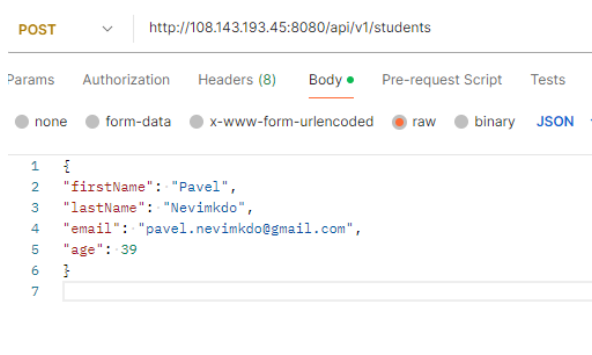
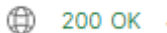
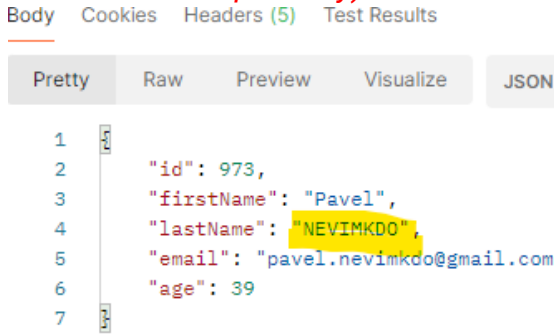
POST:

(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 ",	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	5. v pořádku lze vložit uvedené údaje a	OK

<p>"raw" doplnit tento text:</p> <pre>{ "firstName": "Pavel", "lastName": "Nevimkdo", "email": "pavel.nevimkdo@gmail.com", "age": 39 }</pre> <p>a kliknout na SEND</p>	<p>kliknout na "SEND"</p>	
<p>6. zobrazí se status code 200</p>	<p>6. zobrazí se status code 200</p>	<p>OK</p> 
<p>7. V spodní části BODY se zobrazí doplněné údaje ve formátu JSON a přidělí se ID student</p>	<p>7. v spodní části se zobrazí veškeré zadané údaje a navíc se doplní automaticky ID studenta</p>	<p>BUG:</p> <p><i>Změnila se velikost písma v atributu "lastName" ze zadaného "Nevimkdo" na "NEVIMKDO" (vše velkými písmeny)</i></p> 
<p>8. Následně otevřít aplikaci MySQL</p>	<p>8. Otevřít aplikaci Workbench a připojit se</p>	<p>OK</p>

Workbench - v horní záložce je uvedeno připojení na DTB "engeto_databse_testing"	na uvedenou DTB																			
9. do části Query doplnit tento text (s nově přiděleným ID k novému studentovi): <pre>SELECT * FROM student WHERE id=972 ;</pre> potvrdím CTRL+ENTER	9. lze doplnit uvedené údaje a potvrdit CTRL+ENTER - zobrazí se údaje k studentovi	OK																		
10. Porovnáám 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: <ul style="list-style-type: none">• 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: <i>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.</i> <table><tr><th></th><th>id</th><th>age</th><th>email</th><th>first_name</th><th>last_name</th></tr><tr><td>▶</td><td>972</td><td>39</td><td>pavel.nevimkdo@gmail.com</td><td>Pavel</td><td>NEVIMKDO</td></tr><tr><td>*</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table>		id	age	email	first_name	last_name	▶	972	39	pavel.nevimkdo@gmail.com	Pavel	NEVIMKDO	*	NULL	NULL	NULL	NULL	NULL
	id	age	email	first_name	last_name															
▶	972	39	pavel.nevimkdo@gmail.com	Pavel	NEVIMKDO															
*	NULL	NULL	NULL	NULL	NULL															

Print	
Postman:	MySQL Workbench:

POST

http://108.143.193.45:8080/api/v1/students

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

none

form-data

x-www-form-urlencoded

raw

binary

JSON

```

1 {
2   "firstName": "Pavel",
3   "lastName": "Nevimkdo",
4   "email": "pavel.nevimkdo@gmail.com",
5   "age": 39
6 }

```

body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```

1 {
2   "id": 972,
3   "firstName": "Pavel",
4   "lastName": "NEVIMKDO",
5   "email": "pavel.nevimkdo@gmail.com",
6   "age": 39
7 }

```

1

SELECT

2

*

3

FROM student

4

WHERE id=972

5

;

6

7

result Grid

Filter Rows:

Edit:

id	age	email	first_name	last_name
972	39	pavel.nevimkdo@gmail.com	Pavel	NEVIMKDO
NULL	NULL	NULL	NULL	NULL

2. 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" ,	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: <pre>{ "firstName": " ", "lastName": "Kamos", "email": "jsem.zde.kamo@seznam .cz", "age": 21 }</pre>	5. v pořádku se zapíšou uvedené údaje a kliknout na "SEND"	OK

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

Hodnoty:

id:


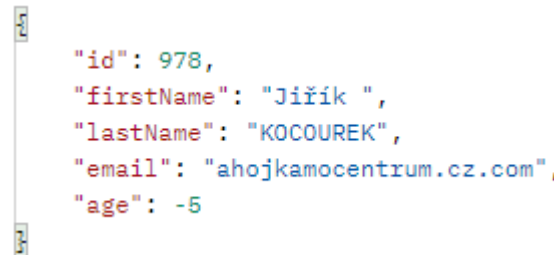
age: -5

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" ,	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 ",	5. v pořádku lze vložit uvedené údaje a kliknout na "SEND"	OK


<pre>"lastName": "Kocourek", "email": "ahojkamocentrum.cz.com", "age": -5 }</pre> <p>a kliknu na SEND</p>		
<p>6. zobrazí se status code 400 - který upozorní, že atribut "age" a "email" mají špatný formát.</p>	<p>6. zobrazí se error status code 400 s informací, že atribut "age" a "email" jsou ve špatném formátu.</p>	<p>BUG:</p> <p>zobrazí se status code 200</p> 
<p>7. V spodní části BODY se zobrazí doplnění chybové hlášky, že atributy "age" a "email" mají špatný formát a je nutné je vyplnit správně.</p>	<p>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"</p>	<p>BUG:</p> <p>student s nesprávnými formáty atributů "age" a "email" byl uložen a dokonce mu bylo přiděleno ID</p> 
<p>8. Postman studenta neuloží, dokud</p>	<p>8. Postman tohoto studenta neuloží,</p>	<p>BUG:</p>

nebudou vyplněné všechny atributy ve správném formátu.	dokud nebudou vyplněné správně všechny atributy včetně momentálně špatně zadaného atributu "age" = -5(záporná hodnota) a atributu "email" - chybí @.	<p>Postman studenta uložil s přiděleným novým ID (přestože je špatný formát atributů "age" a "email"). Takovýto student byl následně uložen do DTB v WorkBench:</p> <table><tr><th>id</th><th>age</th><th>email</th><th>first_name</th><th>last_name</th></tr><tr><td>978</td><td>-5</td><td>ahojkamocentrum.cz.com</td><td>Jiřík</td><td>KOCOUREK</td></tr><tr><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table>	id	age	email	first_name	last_name	978	-5	ahojkamocentrum.cz.com	Jiřík	KOCOUREK	NULL	NULL	NULL	NULL	NULL
id	age	email	first_name	last_name													
978	-5	ahojkamocentrum.cz.com	Jiřík	KOCOUREK													
NULL	NULL	NULL	NULL	NULL													

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" ,	3.	Ize 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: <pre>{ "firstName": "Pavel", "lastName": "Nevimkdo", "email": "pavel.nevimkdo@gmail.com", "age": 39 }</pre> 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  200 OK
7.	V spodní části BODY se zobrazí doplnění chybové hlášky, že studenta nelze uložit, protože student s tímto	7.	v spodní části se zobrazí informace, že student s těmito údaji je již založen.	BUG: Postman studenta uloží, přestože je již

DELETE:

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

```
"id": 997,  
"firstName": "Jirka",  
"lastName": "MAZLIK",  
"email": "jiri.mazlik@gmail.com",  
"age": 15
```


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/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	 Status: 200 OK
5. v části BODY v dolní části nejsou	5. v části BODY nejsou žádné údaje -	OK

	vyplněné žádné údaj - student je smazán	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	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	<div>OK</div> <div><div>1 • SELECT</div><div>2 *</div><div>3 FROM student</div><div>4 WHERE id=997</div><div>5 ;</div><div>6</div><div>7</div></div> <div><div>Result Grid</div><div><div></div><div></div><div></div><div></div><div></div></div><div>Filter Rows: <input type="text"/></div><table><thead><tr><th></th><th>id</th><th>age</th><th>email</th><th>first_name</th><th>last_name</th></tr></thead><tbody><tr><td>*</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></tbody></table></div>		id	age	email	first_name	last_name	*	NULL	NULL	NULL	NULL	NULL
	id	age	email	first_name	last_name										
*	NULL	NULL	NULL	NULL	NULL										

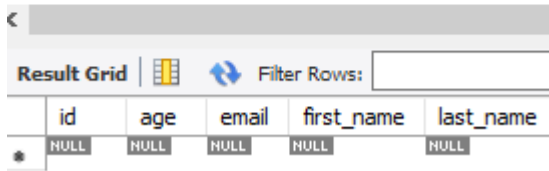
2. DELETE ⇒ Abstrakt: smazání existujícího studenta ID 998 (chybí vyplněný atribut "firstName")

```
"id": 998,
"firstName": " ",
"lastName": "876",
"email": "pavel.nevimkdo_gmail.com",
"age": 1908
```

id	age	email	first_name	last_name
998	1908	pavel.nevimkdo_gmail.com		876
NULL	NULL	NULL	NULL	NULL


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/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  Status: 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

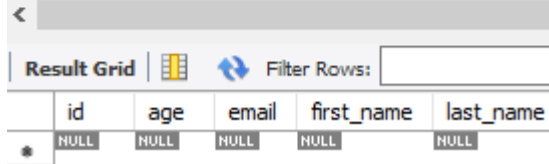
		<div> Body Cookies Headers (4) Test </div> <div> Pretty Raw Preview </div> <div>1</div>
6. Následně otevřít aplikaci MySQL Workbench - v horní záložce je uvedeno připojení na DTB "engeto_databse_testing"	6. aplikace se bez problému otevře s připojením na uvedenou DTB a se záložkou Query s prázdným tělem	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

		<pre> 1 ● SELECT 2 * 3 FROM student 4 WHERE id=998 5 ; 6 7 </pre> 
--	--	---

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:	3. lze vybrat variantu DELETE a vložit uvedené URLa kliknout na tlačítko SEND	OK

"http://108.143.193.45:8080/api/v1/students/1234" , - kliknout na tlačítko SEND		
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  Status: 500 Internal Server Error
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: <pre> "timestamp": "2024-05-08T19:24:01.452+00:00" "status": 500, "error": "Internal Server Error", "message": "", "path": "/api/v1/students/1234" </pre>
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	7. Vložit uvedené příkaz a spustit jej prostřednictvím kombinace kláves CTRL+ enter.	OK

<p>WHERE id=1234 ; a spustím požadavek prostřednictvím CTRL+enter</p>		
<p>8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují a tudíž jej nelze smazat</p>	<p>8. v Workbench se nezobrazí žádné údaje - student s tímto ID neexistují a tudíž jej nelze smazat</p>	<p>OK</p> <pre> 1 • SELECT 2 * 3 FROM student 4 WHERE id=1234 5 ; 6 7 </pre>  <p>The screenshot shows the SQL Workbench interface. At the top, the text 'OK' is displayed in green. Below it, a SQL query is shown with line numbers 1 through 7. The query is: 1 SELECT 2 * 3 FROM student 4 WHERE id=1234 5 ; 6 7 Below the query, there is a 'Result Grid' tab. It shows a table with 5 columns: id, age, email, first_name, and last_name. All cells in the first row are NULL.</p>

4. DELETE ⇒ Abstrakt: smazání studenta s neexistujícím ID s použitím speciálních znaků (podtržítka) 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/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 formát atributu ID	OK  Status: 400 Bad Request
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". <pre>"timestamp": "2024-05-17T13:53:19.260+00", "status": 400, "error": "Bad Request", "message": "ID is not a number", "path": "/api/v1/students/99_8"</pre> -
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/ 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

		Status: 500 Internal Server Error 1
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."	<p>BUG:</p> <ul style="list-style-type: none"> - chybí message - "Špatný formát položky ID, položka nesmí obsahovat mezery". <pre> "timestamp": "2024-05-17T14:07:15.703+00:00", "status": 500, "error": "Internal Server Error", "message": "", "path": "/api/v1/students/%20%20998" </pre>
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	<p>BUG:</p> <ul style="list-style-type: none"> - požadavek proběhne, neobjeví se error status - WorkBench nevidí mezery v příkazu a vezme pouze čísla, důležité zajistit při zakládání nových dat v Postmanu

		<div>kontrolu, že je ID ve správném formátu bez mezer, znaků atd.</div> <div><div>1 • SELECT</div><div>2 *</div><div>3 FROM student</div><div>4 WHERE id= 998</div><div>5 ;</div><div>6</div><div>7</div></div> <div><div><</div><div>Result Grid<div><div><div></div></div><div>Filter Rows:<div></div></div></div><table><tr><th></th><th>id</th><th>age</th><th>email</th><th>first_name</th><th>last_name</th></tr><tr><td>*</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr></table></div></div>		id	age	email	first_name	last_name	*	NULL	NULL	NULL	NULL	NULL
	id	age	email	first_name	last_name									
*	NULL	NULL	NULL	NULL	NULL									

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 "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 uloženy 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.