

# FINÁLNÍ PROJEKT

## č.1



# ENGETO

Autor: Jaroslav Vařeka  
Datum: 8.7.2024

# OBSAH

<b>ZADÁNÍ</b>	<b>3</b>
Přístupové údaje:	3
Poznámky:	3
<b>TESTOVACÍ SCÉNÁŘE</b>	<b>4</b>
<b>EXEKUCE TESTŮ</b>	<b>8</b>
<b>BUG REPORT</b>	<b>15</b>

# ZADÁNÍ

Cílem finálního projektu je otestovat funkčnost aplikace, která slouží k manipulaci s daty o studentech. Aplikace má rozhraní REST-API, které umožňuje vytvoření, smazání a získání dat..

## Přístupové údaje:

Databáze	database: qa_demo Host: aws.connect.psdb.cloud Username: ***** Password: *****
REST-API	http://108.143.193.45:8080/api/v1/students/

## Poznámky:

Nezapomeňte, že v IT se data musí někde uložit a poté získat. Proto ověřte, že data jsou správně uložena a získávána z databáze.

Nezapomeňte do testovacích scénářů uvést testovací data, očekávaný výsledek včetně těla odpovědi a stavových kódů.

# TESTOVACÍ SCÉNÁŘE

*Na základě uvedených testovacích scénářů jsem ověřil(a) funkčnost aplikace.*

## **Abstract: Get data about existing student**

Test data preparation:

Existing student: id=408

```
select * from student where id=408;  
408  32  jan.novak@test.com  Jan  NOVÁK
```

Steps:

- 1) GET <http://108.143.193.45:8080/api/v1/students/408>

Expected results:

- 1) status code 200 OK
- 2) response  
{  
 id: 408,  
 firstName: "Jan",  
 lastName: "NOVÁK",  
 email: "jan.novak@test.com",  
 age: "32",  
}

## **Abstract: Get data about non-existing student**

Test data preparation:

Existing student: id=214

```
select * from student where id=214;  
""
```

Steps:

- 1) GET <http://108.143.193.45:8080/api/v1/students/214>

Expected results:

- 1) status code 404

**Abstract: Delete data about existing student**

Test data preparation:

Existing student: id=355

```
select * from student where id=355;  
355  3      john.doe@gmail.com      John  DOE
```

Steps:

- 1) DELETE <http://108.143.193.45:8080/api/v1/students/355>

Expected results:

- 1) status code 200
- 2) all data about this student are removed

**Abstract: Delete data about non-existing student**

Test data preparation:

Existing student: id=2

```
select * from student where id=2;  
""
```

Steps:

- 1) DELETE <http://108.143.193.45:8080/api/v1/students/2>

Expected results:

- 1) status code 404

**Abstract: Add new student**

Test data preparation:

firstName: Karel, lastName: Pott, age: 19, email: KP42@testmail.com

Steps:

- 1) POST <http://108.143.193.45:8080/api/v1/students/>
- 2) add information about student, body JSON

```
{  
  "firstName": "Karel",  
  "lastName": "Pott",  
  "email": "KP42@testmail.com",  
  "age": 19  
}
```

Expected results:

- 1) status code 201
- 2) response:

```
{
  "id":,
  "firstName": "",
  "lastName": "",
  "email": "",
  "age":
}
```
- 3) database check by id

### **Abstract: Add new student without firstName**

Test data preparation:

lastName: Král, age: 24, email: k4al2@testmail.com

Steps:

- 1) POST `http://108.143.193.45:8080/api/v1/students/`
- 2) add information about student, body JSON

```
{
  "firstName": "",
  "lastName": "Král",
  "email": "k4al2@testmail.com",
  "age": 24
}
```

Expected results:

- 1) status code 400

### **Abstract: Add a new student with a short name**

Test data preparation:

firstName: Bo lastName: Tal, age: 17, email: bo32@testmail.com

Steps:

- 1) POST `http://108.143.193.45:8080/api/v1/students/`
- 2) add information about student, body JSON

```
{  
  "firstName": "Bo",  
  "lastName": "Tal",  
  "email": "bo32@testmail.com",  
  "age": 17  
}
```

Expected results:

- 1) status code 400

# EXEKUCE TESTŮ

Testovací scénáře jsem provedl(a), přikládám výsledky testů.

Get data about existing student

The screenshot displays a REST client interface with the following components:

- URL Bar:** Shows the URL `http://108.143.193.45:8080/api/v1/students/408` with a "Save" button on the right.
- Method and URL:** A dropdown menu shows the method `GET` and the URL `http://108.143.193.45:8080/api/v1/students/408`. A "Send" button is to the right.
- Tabs:** A row of tabs includes "Params", "Auth", "Headers (6)", "Body", "Pre-req.", "Tests", "Settings", and "Cookies". The "Params" tab is currently selected.
- Query Params Table:**

	Key	Value	Bulk Edit
	Key	Value	
- Response Section:**
  - Status Bar:** Shows a globe icon, status `200 OK`, time `130 ms`, size `250 B`, and a "Save Response" button.
  - Format Tabs:** Includes "Pretty", "Raw", "Preview", "Visualize", and "JSON". The "Pretty" tab is selected.
  - Response Body:**

```
1 {
2   "id": 408,
3   "firstName": "Jan",
4   "lastName": "NOVÁK",
5   "email": "jan.novak@test.com",
6   "age": 32
7 }
```



Get data about non-existing student

HTTP

http://108.143.193.45:8080/api/v1/students/214

Save

GET

http://108.143.193.45:8080/api/v1/students/214

Send

Params

Auth

Headers (6)

Body

Pre-req.

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

500 Internal Server Error

169 ms

287 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

"timestamp": "2024-06-30T12:07:15.087+00:00",


"status": 500,


"error": "Internal Server Error",

"message": "",

"path": "/api/v1/students/214"

Delete data about existing student

 **http://108.143.193.45:8080/api/v1/students/355**

 Save

DELETE

▼

http://108.143.193.45:8080/api/v1/students/355

Send

▼

Params

Auth

Headers (6)

Body

Pre-req.

Tests


Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body ▼

 200 OK 100 ms 123 B 

Save Response ▼


Pretty



Raw

Preview















Visualize

Text ▼



1

         Limit to 1000 rows     

1 • `select * from student where id=355;`

Result Grid

</

Delete data about non-existing student

HTTP

http://108.143.193.45:8080/api/v1/students/2

Save

DELETE

http://108.143.193.45:8080/api/v1/students/2

Send

Params

Auth

Headers (6)

Body

Pre-req.

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

500 Internal Server Error111 ms285 BSave Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

"timestamp": "2024-06-30T13:28:59.200+00:00",

"status": 500,

"error": "Internal Server Error",

"message": "",

"path": "/api/v1/students/2"

## Add new student

The screenshot shows an HTTP client interface with the following details:

- URL:** `http://108.143.193.45:8080/api/v1/students/`
- Method:** `POST`
- Body (JSON):**

```
{  "firstName": "Karel",  "lastName": "Pott",  "email": "KP42@testmail.com",  "age": 19}
```
- Response:** `200 OK`, `192 ms`, `250 B`. The response body (Pretty JSON) is:

```
{  "id": 1320,  "firstName": "Karel",  "lastName": "POTT",  "email": "KP42@testmail.com",  "age": 19}
```

The screenshot shows a database query tool interface with the following details:

- Query:** `select * from student where id=1320;`
- Result Grid:** A table with 6 columns: `id`, `age`, `email`, `first_name`, and `last_name`. The results are as follows:

	id	age	email	first_name	last_name
▶	1320	19	KP42@testmail.com	Karel	POTT
*	NULL	NULL	NULL	NULL	NULL

Add new student without firstName

The image shows two screenshots. The top screenshot is from an HTTP client (Postman) showing a POST request to `http://108.143.193.45:8080/api/v1/students/`. The request body is a JSON object: `{ "firstName": "", "lastName": "Král", "email": "k4al2@testmail.com", "age": 24 }`. The response is a 200 OK status with a response time of 196 ms and a body size of 247 B. The response body is shown in a pretty-printed JSON format: `{ "id": 1321, "firstName": "", "lastName": "KRÁL", "email": "k4al2@testmail.com", "age": 24 }`.

The bottom screenshot is from a database query tool (DBeaver) showing a SQL query: `select * from student where id=1321;`. The query results are displayed in a table with 5 columns: `id`, `age`, `email`, `first_name`, and `last_name`. The results show a single row with the following values: `id` is 1321, `age` is 24, `email` is k4al2@testmail.com, `first_name` is NULL, and `last_name` is KRÁL.

id	age	email	first_name	last_name
1321	24	k4al2@testmail.com		KRÁL
*	NULL	NULL	NULL	NULL

Add a new student with a short name

HTTP <http://108.143.193.45:8080/api/v1/students/> Save

POST <http://108.143.193.45:8080/api/v1/students/> Send

Params Auth Headers (8) **Body** Pre-req. Tests Settings Cookies

raw JSON Beautify

```
1 {
2   "firstName": "Bo",
3   "lastName": "Tal",
4   "email": "bo32@testmail.com",
5   "age": 17
6 }
```

Body 200 OK 160 ms 246 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1323,
3   "firstName": "Bo",
4   "lastName": "TAL",
5   "email": "bo32@testmail.com",
6   "age": 17
7 }
```

Limit to 1000 rows

```
1 • select * from student where id=1323;
```

Result Grid					
Filter Rows:					
Edit:					
Export/Import:					
Wrap Cell Content:					
	id	age	email	first_name	last_name
▶	1323	17	bo32@testmail.com	Bo	TAL
*	NULL	NULL	NULL	NULL	NULL

# BUG REPORT

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

Get data about non-existing student

HTTP

http://108.143.193.45:8080/api/v1/students/214

Save

GET

http://108.143.193.45:8080/api/v1/students/214

Send

Params

Auth

Headers (6)

Body

Pre-req.

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

500 Internal Server Error169 ms287 BSave Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

"timestamp": "2024-06-30T12:07:15.087+00:00",

"status": 500,

"error": "Internal Server Error",

"message": "",

"path": "/api/v1/students/214"

Delete data about non-existing student

HTTP

http://108.143.193.45:8080/api/v1/students/2

Save

DELETE

http://108.143.193.45:8080/api/v1/students/2

Send

Params

Auth

Headers (6)

Body

Pre-req.

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

Body

500 Internal Server Error

111 ms

285 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

"timestamp": "2024-06-30T13:28:59.200+00:00",

"status": 500,

"error": "Internal Server Error",

"message": "",

"path": "/api/v1/students/2"



## Add new student

The screenshot shows an HTTP client interface with the following details:

- URL:** `http://108.143.193.45:8080/api/v1/students/`
- Method:** `POST`
- Body (JSON):**

```
{  "firstName": "Karel",  "lastName": "Pott",  "email": "KP42@testmail.com",  "age": 19}
```
- Response (JSON):**

```
{  "id": 1320,  "firstName": "Karel",  "lastName": "POTT",  "email": "KP42@testmail.com",  "age": 19}
```
- Status:** 200 OK, 192 ms, 250 B

The screenshot shows a database query tool interface with the following details:

- Query:** `select * from student where id=1320;`
- Result Grid:**

	id	age	email	first_name	last_name
▶	1320	19	KP42@testmail.com	Karel	POTT
*	NULL	NULL	NULL	NULL	NULL

Add new student without firstName

The image shows two screenshots. The top screenshot is from a REST client (Postman) showing a POST request to `http://108.143.193.45:8080/api/v1/students/`. The request body is a JSON object: `{ "firstName": "", "lastName": "Král", "email": "k4al2@testmail.com", "age": 24 }`. The response is a 200 OK status with a response time of 196 ms and a body size of 247 B. The response body is shown in a pretty-printed JSON format: `{ "id": 1321, "firstName": "", "lastName": "KRÁL", "email": "k4al2@testmail.com", "age": 24 }`.

The bottom screenshot is from a database query tool (DBeaver) showing a SQL query: `select * from student where id=1321;`. The query results are displayed in a table with 5 columns: `id`, `age`, `email`, `first_name`, and `last_name`. The results show a single row with the following values: `id` is 1321, `age` is 24, `email` is k4al2@testmail.com, `first_name` is NULL, and `last_name` is KRÁL.

id	age	email	first_name	last_name
1321	24	k4al2@testmail.com		KRÁL
*	NULL	NULL	NULL	NULL

Add a new student with a short name

HTTP <http://108.143.193.45:8080/api/v1/students/> Save

POST <http://108.143.193.45:8080/api/v1/students/> Send

Params Auth Headers (8) **Body** Pre-req. Tests Settings Cookies

raw JSON Beautify

```
1 {
2   "firstName": "Bo",
3   "lastName": "Tal",
4   "email": "bo32@testmail.com",
5   "age": 17
6 }
```

Body 200 OK 160 ms 246 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1323,
3   "firstName": "Bo",
4   "lastName": "TAL",
5   "email": "bo32@testmail.com",
6   "age": 17
7 }
```

Limit to 1000 rows

```
1 • select * from student where id=1323;
```

Result Grid					
Filter Rows:					
Edit:					
Export/Import:					
Wrap Cell Content:					
	id	age	email	first_name	last_name
▶	1323	17	bo32@testmail.com	Bo	TAL
*	NULL	NULL	NULL	NULL	NULL