

**Министерство науки и высшего образования РФ**  
**Государственное образовательное учреждение высшего**  
**профессионального образования**  
**Белгородский Государственный Технологический Университет им**  
**В. Г. Шухова**

Кафедра программного обеспечения вычислительной техники и  
автоматизированных систем.

Лабораторная работа №5 по дисциплине  
«Технологии Web-программирования»

Тема: REST API

Выполнил:  
студент группы ПВ-41  
Колесников В.Д.  
Проверил: Картамышев С. В.

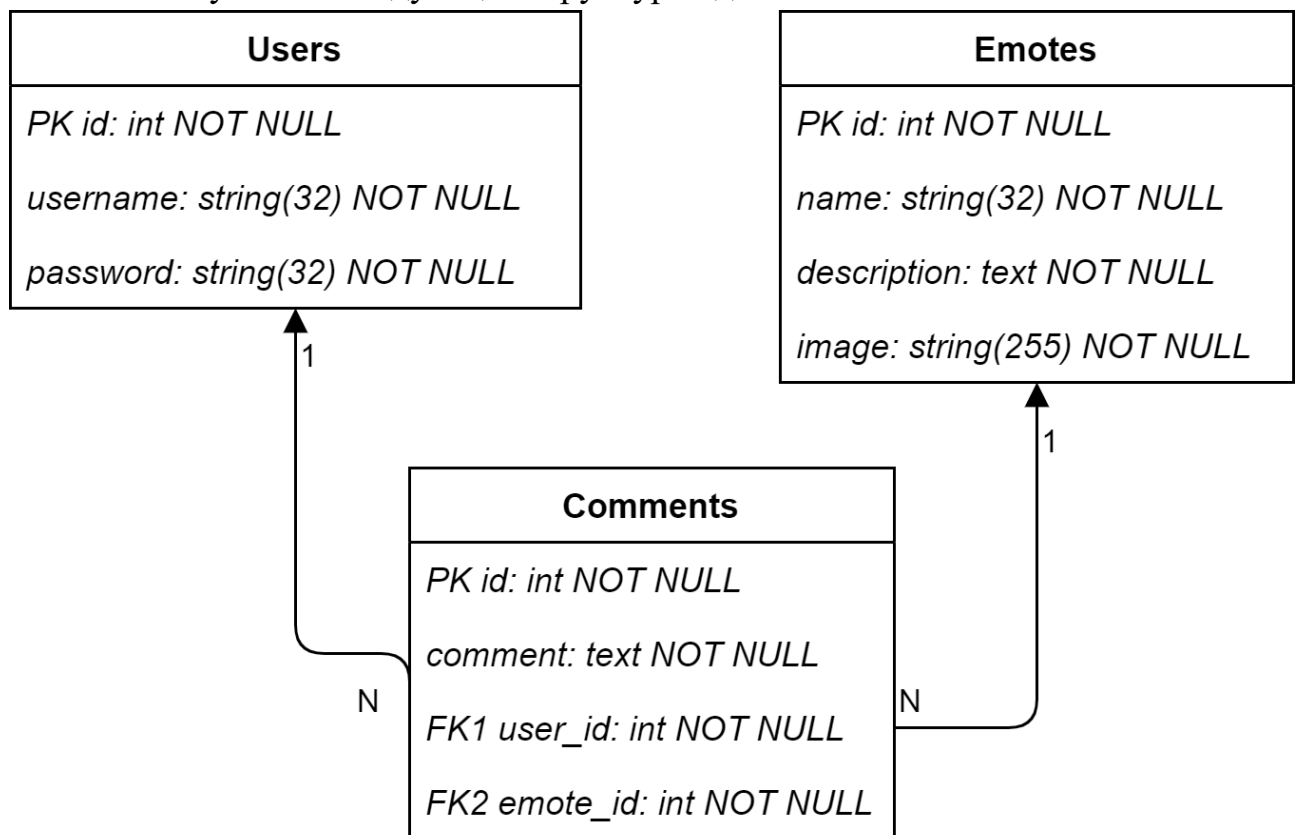
Белгород 2019 г.

**Цель работы:** изучить основы разработки API для web-приложений.

Разработать REST API для своего проекта.

### Выполнение

Имеется получается следующая структура бд:



Для взаимодействия с базой данных было реализовано следующее REST API:

#### User

1. Получение списка всех пользователей

URL: /user/all

Тип запроса: GET

Параметры: нет.

2. Логин

URL: /user/login

Тип запроса: POST

Параметры: username (обязательно, логин) , password (обязательно, пароль).

3. Регистрация

URL: /user/register

Тип запроса: POST

Параметры: username (обязательно, логин) , password (обязательно, пароль).

#### Emote

#### 1. Получение списка всех эмоций

URL: /emote/all

Тип запроса: GET

Параметры: нет.

#### 2. Получение конкретной эмоции

URL: /emote/concrete

Тип запроса: GET

Параметры: id (обязательно, идентификатор объекта).

#### 3. Создание эмоции

URL: /emote/create

Тип запроса: POST

Параметры: принимается JSON-объект с полями: name (обязательно, название эмоции), description (обязательно, описание), image (обязательно, путь\ссылка на изображение).

#### 4. Обновление эмоции

URL: /emote/update

Тип запроса: UPDATE

Параметры: id (обязательно, идентификатор объекта); принимается JSON-объект с полями: name (обязательно, название эмоции), description (обязательно, описание), image (обязательно, путь\ссылка на изображение).

#### 5. Удаление эмоции

URL: /emote/delete

Тип запроса: DELETE

Параметры: id (обязательно, идентификатор объекта).

### **Comment**

#### 1. Получение списка всех комментариев

URL: /comment/all

Тип запроса: GET

Параметры: нет.

#### 2. Получение конкретного комментария по id эмоции, к которой он относится

URL: /comment/emoteid

Тип запроса: GET

Параметры: emote\_id (обязательно, идентификатор эмоции).

3. Получение конкретного комментария по id пользователя, к которому он относится

URL: /comment/userid

Тип запроса: GET

Параметры: user\_id (обязательно, идентификатор пользователя).

4. Создание комментария

URL: /comment/update

Тип запроса: POST

Параметры: принимается JSON-объект с полями: name (обязательно, название эмоции), description (обязательно, описание), image (обязательно, путь\ссылка на изображение).

5. Удаление комментария

URL: /comment/delete

Тип запроса: DELETE

Параметры: id (обязательно, идентификатор объекта).

**Скриншоты**


GET http://127.0.0.1:1199/api/v1/emote/all

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

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

**Body** Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON 

```
1  [
2    {
3      "id": 1,
4      "name": "Okayga",
5      "description": "FeelsOkayMan + Pepega.",
6      "image": "@/assets/images/Okayga.png",
7      "createdAt": "2020-12-05 12:33:23",
8      "updatedAt": "2020-12-05 12:35:19"
9    },
10   {
11     "id": 2,
12     "name": "PeepoClown",
13     "description": "Because twitch viewers are often clowns.",
14     "image": "@/assets/images/PeepoClown.png",
15     "createdAt": "2020-12-05 12:34:01",
16     "updatedAt": "2020-12-05 12:36:14"
17   },
18   {
19     "id": 3,
20     "name": "forsenCD",
21     "description": "D OMEGALUL C emote.",
22     "image": "@/assets/images/forsenCD.jpg",
23     "createdAt": "2020-12-05 12:37:16",
24     "updatedAt": "2020-12-05 12:37:16"
25   },
26   {
27     "id": 4,
28     "name": "FeelsAmazingMan",
29     "description": "When you are totally happy.",
30     "image": "@/assets/images/FeelsAmazingMan.png",
31     "createdAt": "2020-12-05 12:37:47",
32     "updatedAt": "2020-12-05 12:37:47"
33   }
34 ]
```

POST

http://127.0.0.1:1199/api/v1/user/login?username=Pepega&password=forsan

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	username	Pepega
<input checked="" type="checkbox"/>	password	forsan
	Key	Value

Body

Cookies

Headers (10)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1 "Pepega"

**Вывод:** в результате выполнения работы были изучены основы разработки API для web-приложений, и разработано REST API для проекта.