**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.

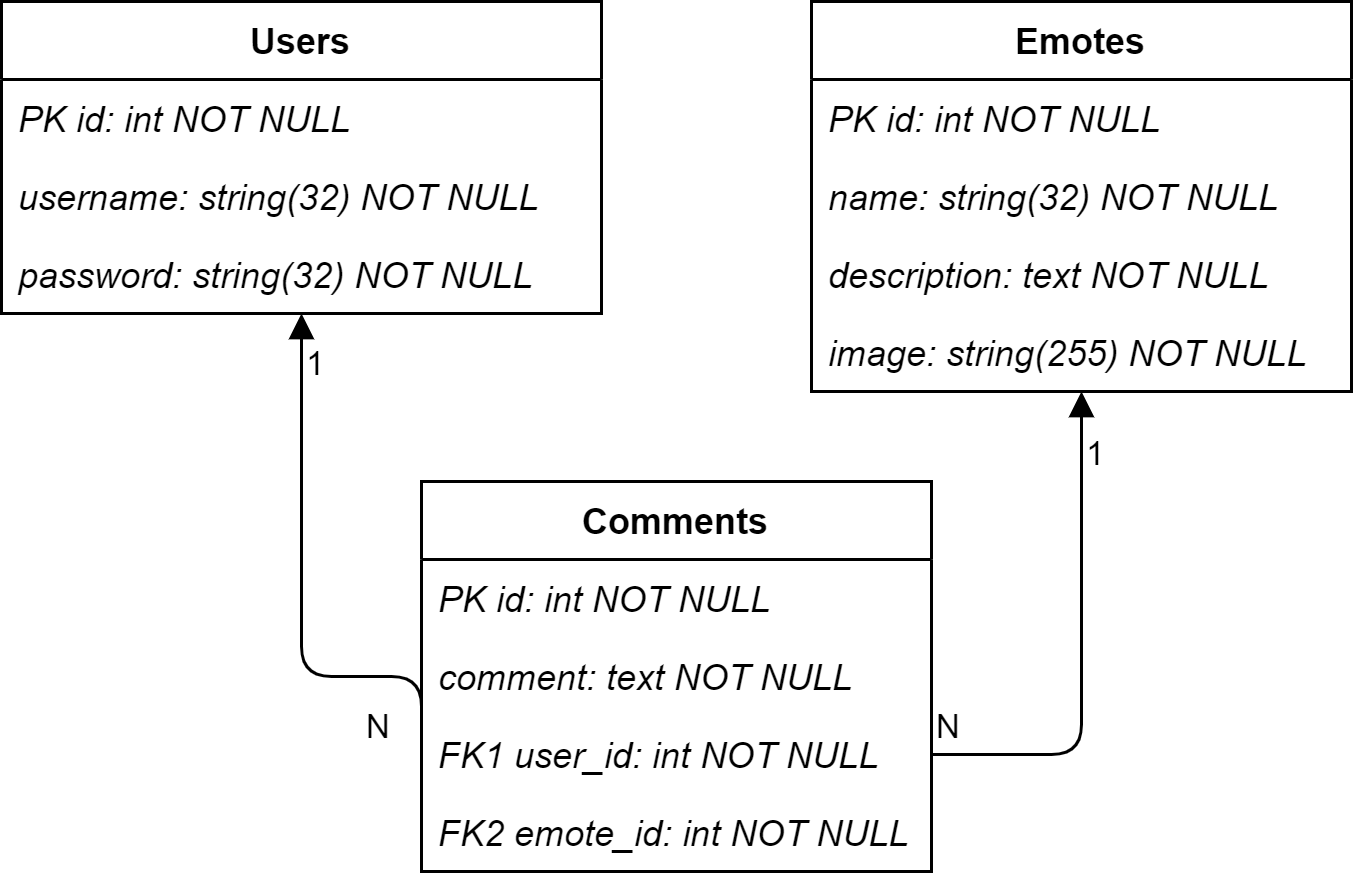
Проверил: Картамышев С. В.

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.

1. Логин

URL: /user/login

Тип запроса: POST

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

1. 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.

1. Получение конкретной эмоции

URL: /emote/concrete

Тип запроса: GET

Параметры: id (обязательно, идентификатор объекта).

1. 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).

1. 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).

1. 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.

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

URL: /comment/emoteid

Тип запроса: GET

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

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

URL: /comment/userid

Тип запроса: GET

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

1. 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).

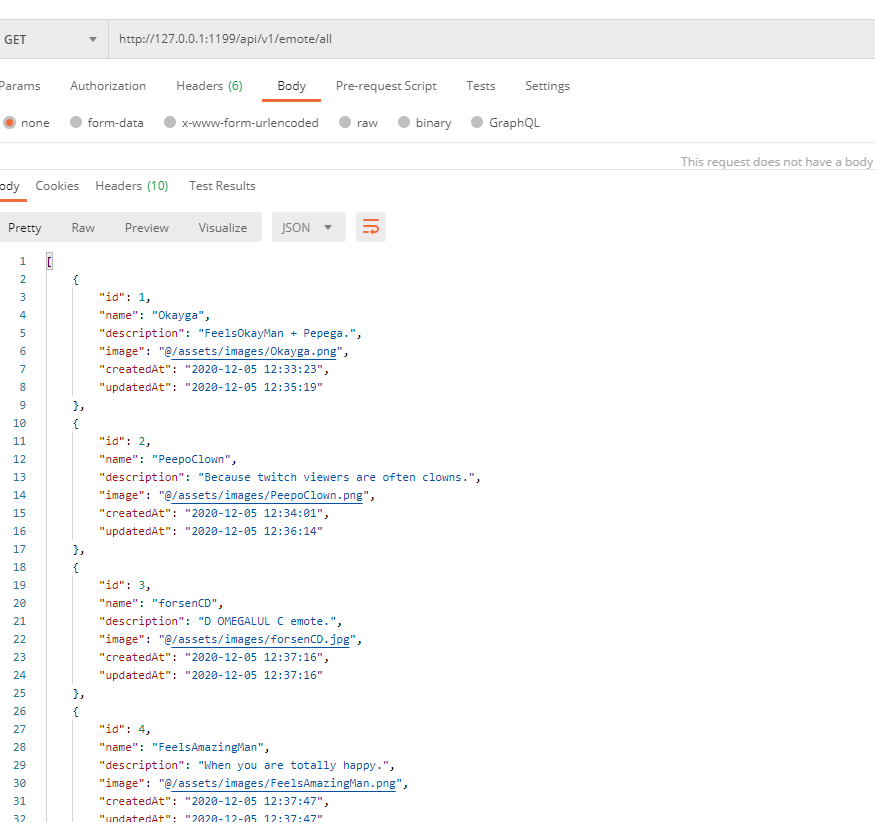
1. Удаление комментария

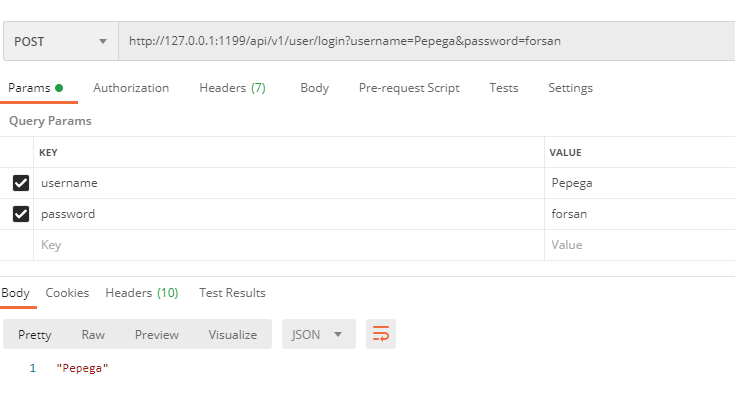
URL: /comment/delete

Тип запроса: DELETE

Параметры: id (обязательно, идентификатор объекта).

**Скриншоты**





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