Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное общеобразовательное учреждение высшего образования

Уфимский университет науки и технологий

Факультет информатики и робототехники

Кафедра вычислительной математики и кибернетики

Отчет к лабораторной работе №4

по дисциплине «Web-технологии»

«Создание и подключение базы данных к веб-приложению. Разработка REST-методов взаимодействия с базой данных веб-приложения»

Выполнили:

Студенты группы МО-321

Шарипов А.Г., Дементьев Д.В.

Проверил:

Юдинцев Б.С.

Уфа-2023

**Задание:**

Создать серверное приложение, используя один из популярных языков программирования: Node.js, Python.

1) Создать базу данных (БД) SQLite с таблицей, которая будет хранить данные, отправляемые с веб-формы вашего приложения. Таблица должна иметь не менее 4-х полей. Процедура создания таблицы должна быть сохранена в отдельный скрипт \*.sql.

2) Подключить БД к приложению и реализовать CRUD-методы для работы с таблицей, т.е. методы должны использовать следующие типы SQL-запросов: SELECT (с фильтром и без), INSERT, UPDATE, DELETE.

3) Разработать REST API и привязать маршруты (routes) вашего вебприложения к соответствующим CRUD-методам. При отправке данных с вашей веб-формы методом POST должны добавляться данные в таблицу, используйте отправку данных на основе подхода AJAX из ЛР №3.

4) Провести тестирование разработанного REST API с помощью любого REST-клиента.

**Теоретическая часть.**

**Основы протокола HTTP.**

HTTP — это [протокол](https://developer.mozilla.org/ru/docs/Glossary/Protocol), позволяющий получать различные ресурсы, например HTML-документы. Протокол HTTP лежит в основе обмена данными в Интернете. HTTP является протоколом клиент-серверного взаимодействия, что означает инициирование запросов к серверу самим получателем, обычно веб-браузером (web-browser).

Клиенты и серверы взаимодействуют, обмениваясь одиночными сообщениями (а не потоком данных). Сообщения, отправленные клиентом, обычно веб-браузером, называются *запросами*, а сообщения, отправленные сервером, называются *ответами*.

Каждый запрос (англ. *request*) отправляется серверу, который обрабатывает его и возвращает ответ (англ. *response*). Между этими запросами и ответами как правило существуют многочисленные посредники, называемые [прокси](https://developer.mozilla.org/ru/docs/Glossary/Proxy_server), которые выполняют различные операции и работают как шлюзы или [кэш](https://developer.mozilla.org/ru/docs/Glossary/Cache), например.

**Основные HTTP-методы.**

GET – метод запрашивает представление ресурса. Запросы с использованием этого метода могут только извлекать данные.

POST – используется для отправки сущностей к определённому ресурсу. Часто вызывает изменение состояния или какие-то побочные эффекты.

PUT – заменяет представление ресурса данными запроса.

DELETE – удаляет указанный ресурс.

**HTTP-коды.**

Код ответа (состояния) HTTP показывает, был ли успешно выполнен определённый HTTP запрос. Коды сгруппированы в 5 классов:

Информационные 100 - 199

Успешные 200 - 299

Перенаправления 300 - 399

Клиентские ошибки 400 - 499

Серверные ошибки 500 – 599

**Ход работы:**

Таблица 1 – Описание REST API веб-приложения

|  |  |  |
| --- | --- | --- |
| Метод HTTP | Действие | Пример URI |
| GET | Получить информацию о всех запросах | <http://127.0.0.1:3000/api/contactrequest/> |
| GET | Получить информацию о запросе по id | [http://127.0.0.1:3000/api/contactrequest/2 (информация](http://127.0.0.1:3000/api/contactrequest/2%20(информация) о запросе №5) |
| GET | Получить информацию о всех запросах, созданных определенным автором | http://127.0.0.1:3000/api/contactrequest/author/Любовь к лабам |
| POST | Создать новый запрос | <http://127.0.0.1:3000/api/contactrequest/> |
| PUT | Обновить запрос | <http://127.0.0.1:3000/api/contactrequest/> |
| DELETE | Удалить запрос | <http://127.0.0.1:3000/api/contactrequest/> |

**1)Получить информацию по id**

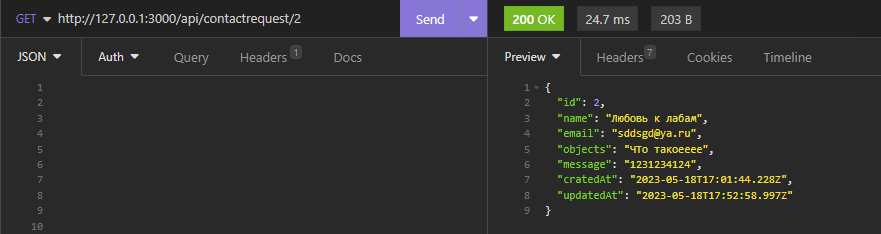


Рис. 1. Получить информацию о запросе по id

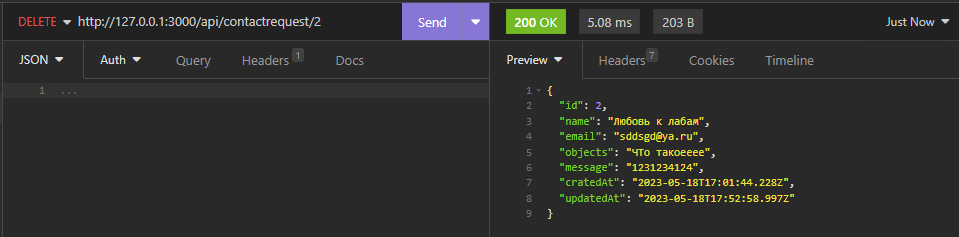
**2)Удаление записи в БД по id**

Рис. 2.1. БД до удаления

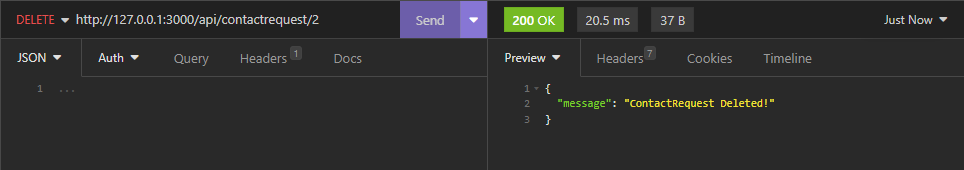


Рис. 2.2. Успешное удаление

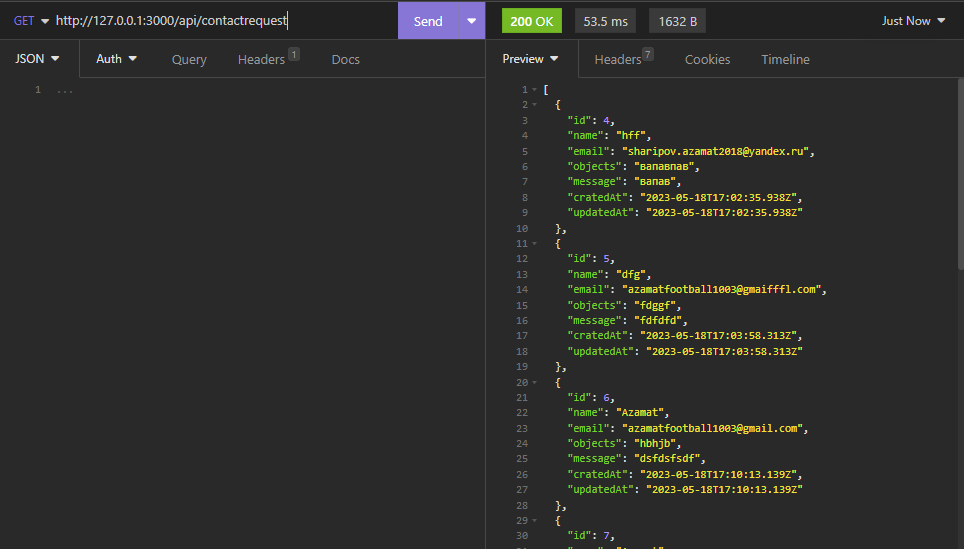


Рис. 2.3. БД после удаления

**3) Получить всю информацию из БД**

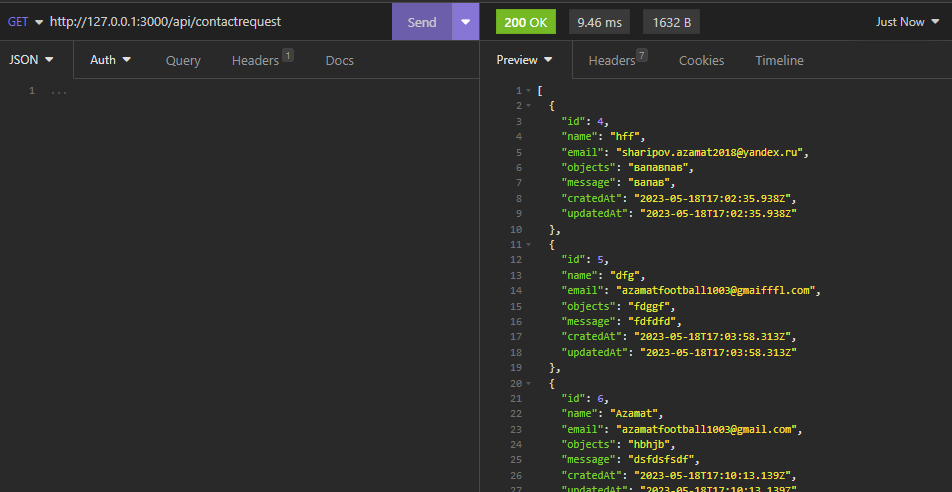
****

Рис. 3. Вся информация в БД

**4) Найти в БД запись по автору**

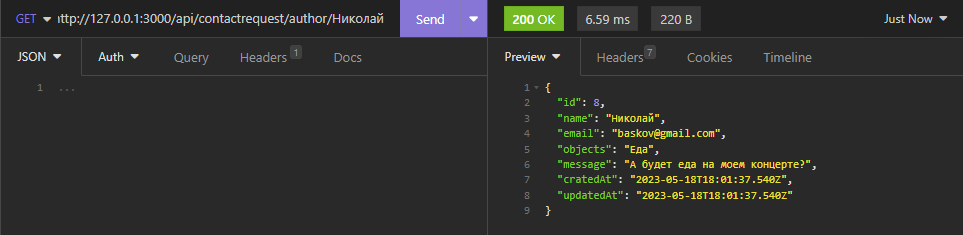
****

Рис. 4. Запись в БД по автору

**5) Обновить запрос**

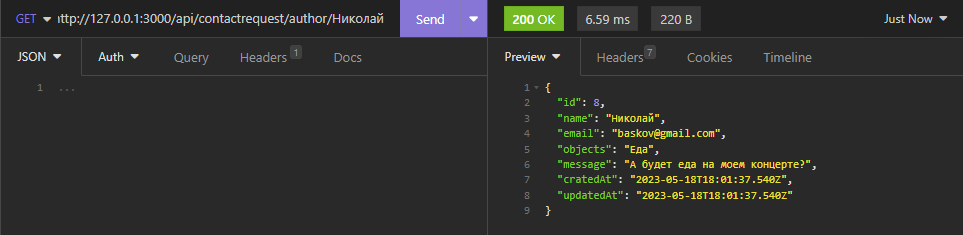
****

Рис. 5.1. Запись до обновления

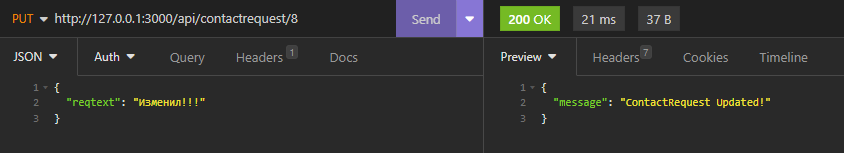
****

Рис. 5.2. Успешное изменение

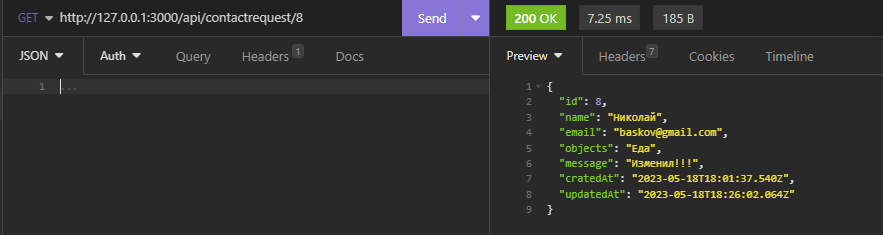
****

Рис. 5.3. Запись после обновления

**Вывод**: в ходе выполнения лабораторной работы создана и подключена БД к приложению, реализованы CRUD-методы, разработан REST API и привязан к CRUD-методам.