

Практическая работа №7. Создание CRUD приложения на React

Цель практической работы: Создать React приложения с базой данных PostgreSQL.

Задачи:

- Реализовать запись и удаление информации из базы данных PostgreSQL в соответствии с полученными запросами HTTP.

Сдача работы:

- Показать выполненное задание во время проведения занятия.
- Ответить на вопросы по практической работе.
- Подготовить и сдать отчет по практической работе.

Теоретические сведения

CRUD - это акроним, который описывает четыре основных операции, которые могут быть выполнены над данными: Создание (Create), Чтение (Read), Обновление (Update) и Удаление (Delete). Для современного разработчика важно знать и понимать CRUD (Create, Read, Update, Delete), поскольку эти операции являются основой большинства приложений и систем, работающих с данными. Так, например, большинство приложений взаимодействуют с базами данных для хранения и извлечения данных. CRUD операции представляют основные операции, выполняемые над данными в базе данных, то есть большинство запросов к базе данных можно имплементировать поверх 4 базовых операций CRUD. Понимание CRUD позволяет разработчику создавать соответствующие запросы, в контексте элементарных операций над данными и обрабатывать данные из базы данных. Или же, многие современные приложения взаимодействуют с удаленными серверами через API. CRUD операции часто используются, также для организации передачи данных между клиентом и сервером, то есть каждому глаголу из базового набора HTTP можно однозначно сопоставить операцию из набора CRUD. Понимание CRUD, в данном контексте, позволяет разработчику правильно использовать API и выполнять соответствующие операции взаимодействия с данными.

Например, рассмотрим пример имплементации базовых бизнес-процессов с операциями CRUD для веб-приложения интернет-магазина. Так операция «Создание» (Create) может относиться к функционалу добавления нового товара в интернет-магазин. Так администратор интернет-магазина может заполнить форму с информацией о товаре, такой как название, описание, цена, изображение и другие характеристики. После заполнения формы и отправки данных, товар добавляется в базу данных интернет-магазина.

Операция «Чтение» (Read) может относиться к функционалу получения информации о товарах из интернет-магазина. Так пользователь может просмотреть список товаров, отфильтровать их по категориям или выполнить поиск по ключевым словам. При выборе конкретного товара, пользователю отображаются подробности о нем, такие как название, описание, цена, изображение и другие характеристики.

Операция «Обновление» (Update) может относиться к функционалу изменения информации о товаре в интернет-магазине. Так пользователь может открыть форму редактирования товара, где может изменить название, описание, цену или другие характеристики товара. После внесения изменений и отправки данных, обновленная информация о товаре сохраняется в базе данных интернет-магазина.

Операция «Удаление» (Delete) Удаление относится к функционалу удаления товара из интернет-магазина. Так пользователь может выбрать определенный товар и запросить его удаление. После подтверждения удаления, товар удаляется из базы данных интернет-магазина и больше не отображается в списке товаров.

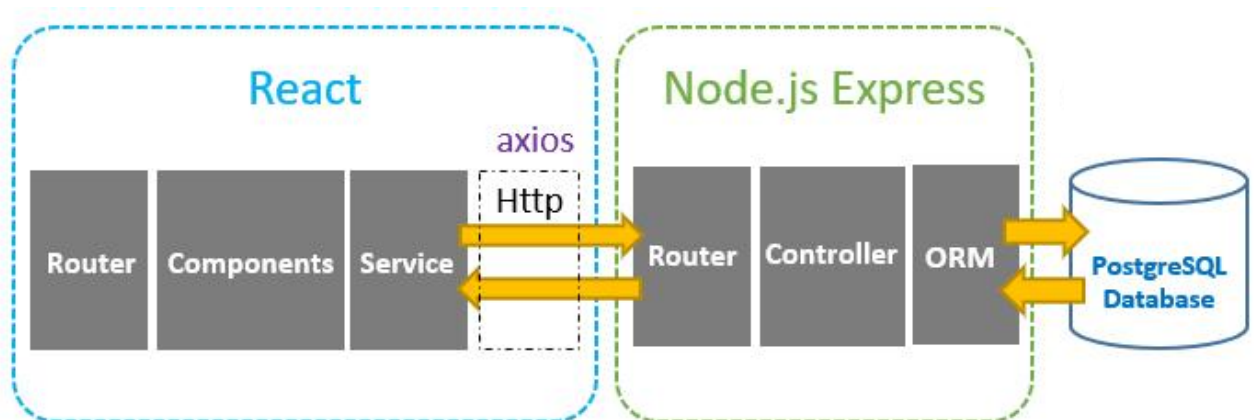


Рисунок 7.1. CRUD архитектура React, NodeJS, PostgreSQL.

Node.js Express экспортирует REST API и взаимодействует с базой данных PostgreSQL, используя Sequelize ORM.

React Client отправляет HTTP-запросы и получает HTTP-ответы с помощью Axios, использует данные о компонентах.

React Router используется для навигации по страницам.

Для связи React приложения с базой данных PostgreSQL, необходимо создать сервер API, который может обрабатывать HTTP-запросы. Создайте

новый каталог и установить новый пакет npm можно с помощью следующих команд, представленных в листинге 7.1.

Листинг 7.1 - Установка пакета

```
mkdir node-postgres && cd node-postgres
npm init
npm i express pg
```

Express - это минималистичный веб-фреймворк, который вы можете использовать для написания веб-приложений на основе технологии Node.js, а клиентская библиотека *node-postgres(pg)*, позволяющая приложениям Node.js взаимодействовать с PostgreSQL.

Дополнительный материал

Документация node-postgres - <https://node-postgres.com> (дата обращения: 01.02.2022).

Getting started with Postgres in your React app - <https://blog.logrocket.com/getting-started-with-postgres-in-your-react-app/> (дата обращения: 01.02.2022).

Authentication with an Express API and Postgres - <https://gist.github.com/laurenfazah/f9343ae8577999d301334fc68179b485> (дата обращения: 01.02.2022).

How to use AccessControl for RBAC and ABAC in Node.js - <https://blog.logrocket.com/using-accesscontrol-for-rbac-and-abac-in-node-js/> (дата обращения: 01.02.2022).

react-abac - <https://www.npmjs.com/package/react-abac> (дата обращения: 01.02.2022).

Node.js JWT Authentication with PostgreSQL example - <https://www.bezkoder.com/node-js-jwt-authentication-postgresql/> (дата обращения: 01.02.2022).

Implementing Role-Based Access Control in a Node.js application - <https://soshace.com/implementing-role-based-access-control-in-a-node-js-application/> (дата обращения: 01.02.2022).