

РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА НА ОСНОВЕ ВЕБ-ИНТЕРФЕЙСА TELEGRAM БОТА

К. О. Коваленко

kiril.kovalenko.79@gmail.com;

Научный руководитель — Ю. А. Бондаренко, старший преподаватель

Интернет-магазины в Telegram предоставляют удобную возможность продажи товаров и услуг внутри мессенджера. Они позволяют создавать каталоги с описаниями и ценами, оформлять заказы и оплачивать их прямо в Telegram. Пользователи могут получать уведомления о новых поступлениях и акциях, взаимодействовать с продавцом через чат-бот, а также использовать интеграции с другими сервисами для упрощения управления магазином. Это особенно полезно для небольших бизнесов, которым необходим простой и эффективный способ продажи онлайн. В данной работе была реализована лишь малая часть всех возможностей мессенджера: взаимодействие пользователей с чат-ботом, просмотр списка товаров, а также деталей каждого по отдельности, оформление заказа и оповещение об успехе. Для осуществления такой функциональности был разработан веб-интерфейс при помощи библиотеки React [1], RESTAPI на основе фреймворка NestJS [2], включенного в него Telegraf модуля для взаимодействия с Telegram API [3].

Ключевые слова: интернет-магазин; веб-интерфейс; React; NestJS; TelegramAPI; чат-бот; MongoDB.

ВВЕДЕНИЕ

Встроенные веб-приложения в Telegram ботах позволяют разработчикам создавать динамические и интерактивные интерфейсы для взаимодействия с пользователями. Они могут добавлять функциональность, используя JavaScript, что открывает широкие возможности для создания различных приложений в Telegram, включая опросы, формы заполнения, каталоги товаров, бронирование услуг и многое другое, также могут использовать AJAX-запросы для обмена данными с сервером без перезагрузки страницы, что позволяет создавать более динамические и отзывчивые интерфейсы. Веб-приложения могут быть интегрированы с помощью API Telegram, который предоставляет различные методы для взаимодействия с ботами, такие как отправка и получение сообщений, обработка команд, сохранение данных и многое другое. Они также могут использовать эти методы для взаимодействия с пользователями и обеспечения нужной функциональности. Такой подход крайне удобен, поскольку пользоваться веб-приложениями

чата, без необходимости переключаться на отдельные веб-страницы или приложения.

ОБЗОР ТЕХНОЛОГИЙ

Выбор инструмента для создания клиентской части приложения пал на React – это популярная JavaScript-библиотека, используемая для разработки пользовательского интерфейса веб-приложений. Ее удобство проявляется в следующем: основана на компонентном подходе, применяет однонаправленный поток данных, может быть дополнена другими инструментами и библиотеками.

Оптимальным вариантом для создания серверной части стал NestJS–фреймворк для разработки серверных приложений на TypeScript. Он основан на архитектурном стиле MVC (Model - View - Controller) и принципах Dependency Injection, что обеспечивает легкость в организации кода и управлении зависимостями. Модульность является важной частью NestJS, позволяя разделять функциональность и повторно использовать код. Фреймворк поддерживает TypeScript, что обеспечивает статическую типизацию и улучшает надежность кода.

В качестве базы данных использовалась документоориентированная MongoDB – высокопроизводительная и гибкая система управления базами данных, обеспечивающая масштабируемость, поддержку индексов, возможности запросов, репликацию, горизонтальное разделение данных (шардинг), поддержку различных языков программирования и богатую экосистему инструментов. Она удобна для разработчиков, позволяя легко адаптировать данные, обеспечивать эффективный доступ к ним, обрабатывать большие объемы информации.

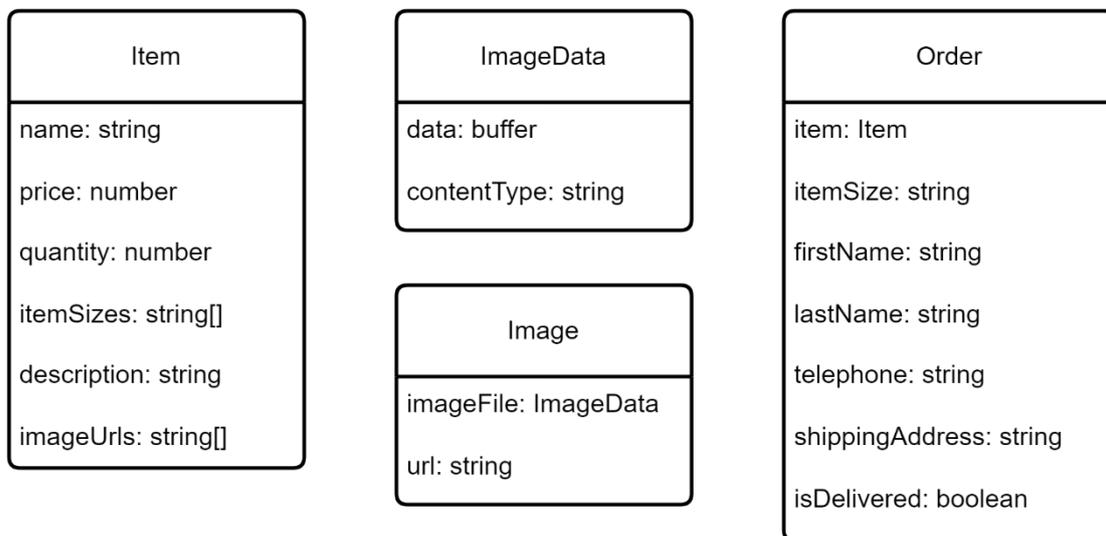
В дополнение к этому, приложение интегрировано в широко используемый мессенджер Telegram для удобного доступа к интернет-магазину. Взаимодействие осуществляется посредством Telegram API, который требует регистрации чат-бота через BotFather и получения уникального API-токена для последующей работы.

СХЕМЫ СУЩНОСТЕЙ БАЗЫ ДАННЫХ

Так как используемой базой данных является документоориентированная MongoDB, то данные хранятся в виде JSON (JavaScript Object Notation) объектов, схемы которых представлены на рисунке.

Покупаемой единицей в интернет-магазине является «Item». Изображения товаров хранятся в базе данных в формате Base64, поэтому для упрощения нужны две сущности: «ImageData», в поле *data* которой помещена картинка в формате *buffer*, и «Image», в поле *imageFile* которой

вложен документ «ImageData». В соответствии со схемой «Order», в которую вложен «Item», хранятся заказы.



Схемы сущностей базы данных

КЛИЕНТСКАЯ ЧАСТЬ

Были определены пути и страницы, которые по ним открываются: «MarketPage» по пути «/», «ItemDetailsPage» по пути «/item-details/:id», «OrderPage» по пути «/order/:id», где вместо параметра *id* в обоих случаях подставляется уникальный идентификатор (*id*) товара.

При загрузке страниц отправляются запросы на сервер для получения данных о товарах, так как эта процедура повторяется достаточно часто, был написан hook «useFetch» (функция параметром принимающая *url*, то есть адрес, по которому будет отправляться GET запрос и возвращающая преобразованный в JSON объект ответ от сервера), который переиспользуется в разных компонентах.

При помощи props (сокращение от «properties», механизм передачи данных от одного компонента к другому) данные попадают, например, в компонент «Item» и далее отображаются на странице.

Таким образом веб-приложения собирается в единое целое, образуя иерархическую структуру. Это позволяет создавать сложные пользовательские интерфейсы, разбивая их на более мелкие и управляемые компоненты.

СЕРВЕРНАЯ ЧАСТЬ

Задачи сервера на NestJS: валидация, маршрутизация и обработка HTTP-запросов, формирование соответствующих HTTP-ответов, интеграция с базой данных, инициализация Telegram бота.

Так как серверное приложение имеет модульную структуру, то все части необходимо собрать в AppModule (главный модуль, который определяет структуру и зависимости приложения): с помощью пакета Telegraf и токена инициализирован бот, пакета Mongoose подключен удаленный кластер базы данных MongoDB, также Items Module, Image Module, Orders Module и Bot Update (модуль, где прописаны шаблоны, отвечающие за взаимодействие с ботом).

Секретные переменные окружения хранятся в .env файле и подтягиваются при сборке проекта, чтобы злоумышленники не смогли легко получить доступ к ним.

Результатом разработки серверной части является REST API, предоставленный для клиентов, а также чат-бот и шаблоны для взаимодействия с ним.

ВЫВОДЫ

Интернет-магазин, разработанный на основе веб-интерфейса Telegram ботас использованием React, NestJS и Telegram API, предоставляет удобную и интегрированную платформу для пользователей, где они могут заказывать товары и взаимодействовать с магазином, не покидая Telegram. Интеграция с Telegram API обеспечивает возможность взаимодействия через интерфейс чат-бота, а React и NestJS предоставляют реактивность, безопасность и простоту разработки как клиентской, так и серверной частей. Этот проект является результатом эффективного использования современных технологий и может быть дальше расширен и развит для улучшения функциональности и удобства пользователей.

Библиографические ссылки

1. React – JavaScript-библиотека для создания пользовательских интерфейсов [Электронный ресурс]. URL: <https://ru.legacy.reactjs.org> (дата обращения: 30.05.2023).
2. NestJS – a progressive Node.js framework [Электронный ресурс]. URL: <https://nestjs.com> (дата обращения: 31.05.2023).
3. Telegram API docs [Электронный ресурс]. URL: <https://core.telegram.org/bots> (дата обращения: 03.06.2023).