

АРХИТЕКТУРА ИНТЕГРАЦИИ TELEGRAM-БОТА С УЧЁТНОЙ СИСТЕМОЙ 1С ЧЕРЕЗ HTTP-СЕРВИСЫ

ARCHITECTURE OF TELEGRAM-BOT INTEGRATION WITH 1C ACCOUNTING SYSTEM THROUGH HTTP SERVICES

УДК 004. 45

Чернышев Владислав Максимович, магистрант, Белгородский государственный национальный исследовательский университет, г. Белгород
Chernyshev Vladislav Maksimovich, Graduate Student, Belgorod State National Research University, Russia, Belgorod

Аннотация

В статье описывается архитектура интеграции чат-бота, функционирующего в мессенджере Telegram, с корпоративной учётной системой «1С: Предприятие» на основе HTTP-сервисов. Рассматриваются способы организации двустороннего обмена данными, форматы сообщений (JSON), механизмы аутентификации и кэширования. Приводятся примеры реализации HTTP-сервиса на встроенном языке 1С и клиентского модуля на Python. Оцениваются временные характеристики и безопасность предложенного решения. Особое внимание уделено обеспечению безопасности и снижению нагрузки на учётную систему за счёт кэширования; экспериментально подтверждена высокая надёжность и производительность подхода. Предложенная архитектура может быть использована для построения интеграционных решений между Telegram и другими ERP-системами.

Annotation

The article describes the architecture of integrating a chatbot operating in the Telegram messenger with the 1C: Enterprise corporate accounting system based on HTTP services. The ways of organizing two-way data exchange, message formats (JSON), authentication and caching mechanisms are considered. Examples of

implementing an HTTP service in the embedded 1C language and a client module in Python are given. The time characteristics and security of the proposed solution are estimated. Special attention is paid to ensuring security and reducing the load on the account system by caching; the high reliability and performance of the approach have been experimentally confirmed. The proposed architecture can be used to build integration solutions between Telegram and other ERP systems.

Ключевые слова: интеграция, чат-бот, Telegram, 1С: Предприятие, HTTP-сервис, JSON, API, аутентификация, кэширование.

Keywords: integration, chatbot, Telegram, 1С:Enterprise, HTTP service, JSON, API, authentication, caching.

Автоматизация взаимодействия с контрагентами является одной из приоритетных задач при построении эффективного предприятия. Широкое распространение мессенджеров, в частности Telegram, открывает новые возможности для организации быстрого и удобного канала связи с поставщиками и клиентами. Однако для того, чтобы чат-бот мог предоставлять актуальную информацию о заказах, остатках, ценах и других данных, необходима его интеграция с внутренней учётной системой – например, с платформой «1С: Предприятие», которая является стандартом для управления предприятием в России.

На рынке существуют готовые сервисы-конструкторы ботов и модули для 1С, но они часто имеют ограничения по функциональности, безопасности или требуют регулярной абонентской платы. Альтернативой является создание собственного интеграционного решения на базе открытых технологий. Цель данной работы – предложить и обосновать архитектуру интеграции Telegram-бота с 1С, обеспечивающую надёжный, безопасный и масштабируемый обмен данными.

На рисунке 1 представлена архитектура интеграции.

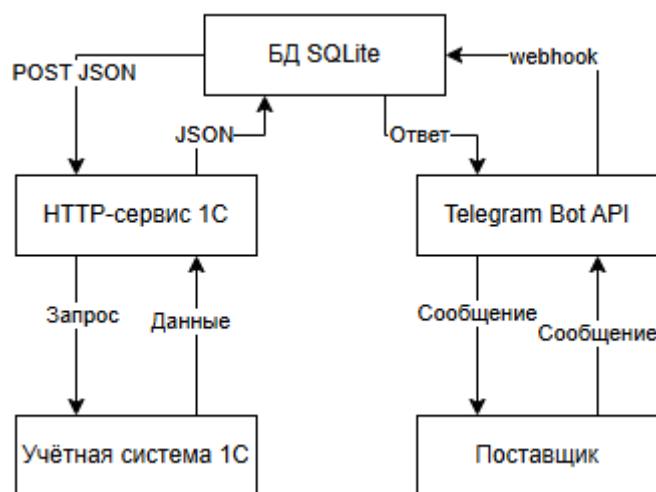


Рисунок 1 – Архитектура интеграции

Предлагаемая архитектура, представленная на рисунке, включает три основных компонента:

1) Telegram-бот (клиент) – программный модуль на Python, взаимодействующий с пользователями и формирующий запросы к 1С.

2) HTTP-сервис – веб-приложение, опубликованное на платформе 1С, которое принимает входящие запросы, обращается к базам данных и возвращает результат.

3) Учётная система 1С – источник данных (заказы, номенклатура, остатки, цены).

Взаимодействие между ботом и 1С осуществляется по протоколу HTTPS. Бот инициирует POST-запросы к HTTP-сервису, передавая параметры в формате JSON. Сервис аутентифицирует запрос (по API-ключу), выполняет логику получения данных и возвращает ответ также в JSON. Для снижения нагрузки на 1С и ускорения ответов на повторные запросы в боте реализован кэш (например, с использованием библиотеки cachetools).

В конфигурации 1С создаётся общий модуль, который экспортируется как HTTP-сервис. Настройка публикации выполняется через веб-сервер (IIS или Apache). Сервис принимает POST-запросы, разбирает JSON, проверяет API-ключ, выполняет запросы к базе данных и формирует ответ.

Бот написан на Python с использованием асинхронной библиотеки aiogram. Основной цикл обработки сообщений включает:

- идентификацию поставщика (по id чата);
- классификацию запроса (см. статью 1);
- формирование HTTP-запроса к сервису 1С;
- обработку ответа и отправку сообщения.

Для повышения производительности реализовано кэширование ответов (например, остатков и цен) на 5 минут. Ключ кэша строится из типа запроса и параметра (артикула). При повторном запросе тех же данных в течение времени жизни кэша бот возвращает сохранённый результат без обращения к 1С.

Для защиты канала связи используется протокол HTTPS (TLS 1.2/1.3). Аутентификация запросов к 1С осуществляется по API-ключу, который передаётся в заголовке X-API-Key. На стороне 1С создан отдельный пользователь с минимальными правами (только чтение справочников и регистров, необходимых для работы). В случае неверного ключа сервис возвращает HTTP 403. Дополнительно рекомендуется ограничить IP-адреса, с которых разрешены запросы к HTTP-сервису (только IP сервера бота).

Экспериментальные испытания на стенде, имитирующем нагрузку 200 запросов в день, показали следующие результаты:

- Среднее время обработки запроса ботом без кэша: 1,2 секунды (включая время запроса к 1С и обработку).
- Среднее время обработки запроса с использованием кэша: 0,3 с.
- Максимальное время ответа при пиковой нагрузке (50 запросов/час) не превысило 2 секунд.
- При недоступности HTTP-сервиса бот после трёх попыток отправляет сообщение о временных неполадках и предлагает обратиться к менеджеру.

Надёжность интеграции обеспечивается за счёт асинхронной архитектуры бота, повторных попыток (retry) при временных сбоях и подробного логирования всех ошибок.

Предложенная архитектура интеграции Telegram-бота с 1С на основе HTTP-сервисов является простой в реализации, безопасной и масштабируемой. Использование открытых стандартов (HTTP, JSON, регулярные выражения) позволяет адаптировать решение для других ERP-систем. Кэширование часто запрашиваемых данных снижает нагрузку на учётную систему и ускоряет ответ. Экспериментальная проверка подтвердила, что среднее время ответа на запрос поставщика составляет 0,3–1,2 секунды, а при сбоях бот корректно уведомляет пользователя. Данная архитектура может быть рекомендована для построения производственных систем автоматизации взаимодействия с контрагентами.

Литература

1. 1С: Предприятие 8.3. Документация разработчика. // Фирма «1С». 2022.
2. Telegram Bot API Documentation [Электронный ресурс]. – Режим доступа: <https://core.telegram.org/bots/api> (дата обращения: 05.01.2026).
3. Python Software Foundation. aiohttp — HTTP client/server for asyncio [Электронный ресурс]. – Режим доступа: <https://docs.aiohttp.org/> (дата обращения: 06.01.2026).
4. Саати Т. Принятие решений. Метод анализа иерархий. // Радио и связь. 1993. – 278 с.
5. Хрусталева Е.Ю. Разработка сложных отчетов в «1С:Предприятии 8». // ООО «1С-Пабблишинг», 2019. – 485 с. (включая создание HTTP-сервисов и внешних обработок).

Literature

1. 1С: Enterprise 8.3. Developer Documentation. // 1С Company. 2022.
2. Telegram Bot API Documentation [Electronic resource]. – Access mode: <https://core.telegram.org/bots/api> (accessed: 05.01.2026).
3. Python Software Foundation. aiohttp — HTTP client/server for asyncio [Electronic resource]. – Access mode: <https://docs.aiohttp.org/> (accessed on 06.01.2026).

4. Saaty T. Decision Making. The Analytic Hierarchy Process. // Radio and Communications. 1993. – 278 p.
5. Khrustaleva E.Yu. Development of Complex Reports in 1C:Enterprise 8. // 1C-Publishing LLC, 2019. – 485 p. (including the creation of HTTP services and external handlers).