

Н. А. Ермилов^{1✉}, Е. Ю. Воронкин¹

Разработка серверной части SaaS-платформы по управлению e-commerce бизнесом

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: nikitaermilovvv@gmail.com

Аннотация. В статье рассматривается процесс разработки серверной части SaaS-платформы для управления e-commerce бизнесом и автоматизации взаимодействия между самозанятыми продавцами и партнерскими магазинами, размещающими товары на маркетплейсах. Актуальность работы обусловлена ростом числа независимых продавцов, реализующих товары через социальные сети и интернет-магазины, но не имеющих собственной инфраструктуры для работы с маркетплейсами. Разработанная система предоставляет возможность публикации товаров, управления заказами, формирования документов для отгрузки, а также интеграции с внешними сервисами и сайтами. Серверная часть реализована с использованием современных веб-технологий и ориентирована на масштабируемость, безопасность и централизованное управление бизнес-процессами. В результате создана программная платформа, обеспечивающая автоматизацию логистики, обработки заказов и администрирования e-commerce операций.

Ключевые слова: SAAS, e-commerce, backend, API, автоматизация бизнеса, маркетплейсы, управление заказами, интеграция сервисов, логистика, веб-разработка

N. A. Ermilov^{1✉}, E. Yu. Voronkin¹

Development of the server part of a SaaS-platform for e-commerce business management

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: nikitaermilovvv@gmail.com

Abstract. The article discusses the development process of the backend part of a SaaS platform for e-commerce business management and automation of interactions between self-employed sellers and partner stores operating on marketplaces. The relevance of the work is conditioned by the growing number of independent sellers who sell products through social networks and websites but do not have direct access to marketplace infrastructure. The developed system provides product management, order processing, shipment documentation generation, and integration with external services and websites. The backend architecture is designed with a focus on scalability, security, and centralized business process management. As a result, a software platform was created to automate logistics, order handling, and administrative e-commerce operations.

Keywords: SAAS, e-commerce, backend, API, business automation, marketplaces, order management, logistics, web development

Введение

Современный рынок электронной коммерции активно развивается за счет малого бизнеса, самозанятых и независимых производителей. Значительная

часть продавцов реализует товары через социальные сети, мессенджеры и собственные интернет-магазины, однако сталкивается с ограничениями при выходе на крупные маркетплейсы.

Основными проблемами являются сложность интеграции с инфраструктурой маркетплейсов, высокая стоимость логистики при самостоятельной отправке заказов и отсутствие централизованного инструмента управления продажами.

Для решения данной задачи была разработана SaaS-платформа, позволяющая самозанятым размещать товары через партнерские магазины, работающие на маркетплейсах. Платформа обеспечивает управление каталогом товаров, обработку заказов, формирование документов для отгрузки и контроль логистических процессов.

Целью работы является разработка серверной части SaaS-платформы для автоматизации e-commerce бизнес-процессов. Для достижения цели были выполнены следующие задачи:

- анализ требований к платформе;
- проектирование серверной архитектуры;
- разработка API для взаимодействия клиентской части и внешних сервисов;
- реализация системы управления товарами, заказами и пользователями;
- интеграция модулей логистики и документооборота;
- тестирование и развертывание серверной части.

Методы и материалы

Перед началом разработки серверной части SaaS-платформы был проведен анализ предметной области и сформулированы основные требования к системе. Разрабатываемая платформа ориентирована на автоматизацию процессов управления e-commerce бизнесом и взаимодействия между самозанятыми продавцами, партнерскими магазинами и администрацией сервиса. На этапе проектирования были определены ключевые бизнес-процессы платформы, включающие управление товарами, обработку заказов, сопровождение отправок, работу с документацией и интеграцию с внешними сервисами. Также была спроектирована логика разграничения ролей пользователей, предусматривающая различные уровни доступа для продавцов, администраторов и партнерских аккаунтов.

Для реализации серверной части использовался язык программирования Python 3.11 и веб-фреймворк Flask 3, обеспечивающий обработку HTTP-запросов и построение серверного приложения. Архитектура приложения построена по модульному принципу с разделением конфигурации, маршрутов и бизнес-логики, что упрощает поддержку и развитие проекта. Взаимодействие между клиентской и серверной частями реализовано посредством REST API, позволяющего передавать данные между личным кабинетом пользователя, административной панелью и серверной инфраструктурой. Для обеспечения работы приложения в production-среде используется WSGI-сервер Gunicorn, а корректная

обработка запросов при работе за reverse проху реализована с использованием ProxyFix из библиотеки Werkzeug.

Для хранения данных используется реляционная модель базы данных. В процессе локальной разработки и тестирования применялась СУБД SQLite, обеспечивающая простоту настройки и автономность окружения. Для production-развертывания используется PostgreSQL, подключаемая через переменные окружения, что позволяет обеспечить масштабируемость и более надежную работу с данными. Взаимодействие с базой данных реализовано с использованием ORM SQLAlchemy 2, позволяющей работать с сущностями приложения на уровне объектной модели. Для управления версиями структуры базы данных и безопасного внесения изменений применяется система миграций Alembic.

Серверная часть платформы интегрирована с рядом внешних сервисов, необходимых для автоматизации бизнес-процессов. Для выполнения HTTP-запросов к внешним API используется библиотека requests. Реализована интеграция с API маркетплейса Wildberries, позволяющая взаимодействовать с сервисами управления товарами и контентом. Дополнительно реализована интеграция с банковской системой Точка, включающая OAuth-авторизацию и обработку входящих вебхуков. Для проверки JWT-подписей используются библиотеки PyJWT и cryptography. Генерация вспомогательных документов, PDF-файлов и обработка изображений выполняется с использованием библиотек ReportLab и Pillow. Для обеспечения воспроизводимости окружения и упрощения развертывания серверной части используется контейнеризация на основе Docker.

Результаты

В результате проведенного анализа и проектирования была реализована серверная часть SaaS-платформы для управления e-commerce бизнесом. На основе спроектированной архитектуры были реализованы основные серверные модули системы, отвечающие за управление пользователями, товарами, заказами и бизнес-процессами платформы. В серверной части реализован модуль управления товарами, позволяющий сохранять, обновлять и обрабатывать данные о продукции пользователей. Также реализована система обработки заказов, включающая получение информации о заказах, обновление их статусов и сопровождение логистических операций.

Для хранения данных были реализованы модели сущностей и связи между ними с использованием SQLAlchemy. Структура базы данных включает сущности пользователей, товаров, заказов, магазинов, отправок и интеграционных данных. При этом с помощью Alembic реализован механизм миграций, позволяющий управлять изменениями структуры базы данных и поддерживать целостность данных при развитии платформы.

Одним из ключевых результатов разработки стала реализация интеграции с API Wildberries и банковской системой Точка. Модуль Wildberries обеспечивает обмен данными с маркетплейсом и автоматизацию операций, связанных с товарами и заказами, в то время как интеграция с банком позволяет обрабатывать финансовые события, OAuth-авторизацию и входящие вебхуки. Для обеспечения

безопасности реализована система аутентификации и разграничения прав доступа пользователей. Серверная часть поддерживает контроль ролей и ограничение доступа к административным функциям платформы.

Дополнительно реализован функционал генерации документов и вспомогательных материалов для логистики с использованием ReportLab и Pillow. В целом разработанная серверная архитектура обладает модульной структурой, что обеспечивает возможность дальнейшего масштабирования и расширения функциональности платформы.

Заключение

В статье рассмотрен процесс разработки серверной части SaaS-платформы для управления e-commerce бизнесом, ориентированной на автоматизацию взаимодействия между самозанятыми продавцами, партнерскими магазинами и административной системой платформы. Основной целью проекта являлось создание масштабируемой и безопасной серверной инфраструктуры, обеспечивающей управление товарами, обработку заказов, сопровождение логистических процессов и централизованное администрирование основных бизнес-операций. В ходе работы были проанализированы требования к системе, определены ключевые бизнес-процессы и спроектирована архитектура серверного приложения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Docker Inc. Docker Documentation. – Режим доступа: <https://docs.docker.com/> (дата обращения: 27.03.2026). – (Контейнеризация и оркестрация серверного приложения.)
2. PostgreSQL Global Development Group. PostgreSQL 16 Documentation. – Режим доступа: <https://www.postgresql.org/docs/16/> (дата обращения: 25.03.2026). – (Реляционная СУБД для production-среды.)
3. SQLAlchemy Authors. SQLAlchemy 2.0 Documentation. – Режим доступа: <https://docs.sqlalchemy.org/> (дата обращения: 26.03.2026). – (ORM и ядро для работы с базой данных.)
4. Alembic Project. Alembic Migration Documentation. – Режим доступа: <https://alembic.sqlalchemy.org/> (дата обращения: 26.03.2026). – (Управление версиями схемы БД.)
5. Gunicorn Project. Gunicorn WSGI Server Documentation. – Режим доступа: <https://docs.gunicorn.org/> (дата обращения: 24.03.2026). – (Production-сервер для WSGI-приложений.)
6. Wildberries. API Marketplace (v2). – Режим доступа: <https://openapi.wildberries.ru/> (дата обращения: 20.03.2026). – (Интеграция с маркетплейсом: товары, заказы, логистика.)
7. Банк «Точка». API для партнеров. – Режим доступа: <https://enter.tochka.com/api> (дата обращения: 22.03.2026). – (Официальная документация на русском языке, OAuth 2.0 и вебхуки.)
8. Python Software Foundation. The Python Standard Library: requests, PyJWT, cryptography, hashlib. – Режим доступа: <https://docs.python.org/3/library/> (дата обращения: 27.03.2026). – (Стандартные и дополнительные библиотеки для HTTP, JWT, шифрования.)

© Н. А. Ермилов, Е. Ю. Воронкин, 2026