

Санкт-Петербургский государственный университет

Группа 21.Б10-мм

***КСЕНЧИК Никита Владимирович***

Выпускная квалификационная работа

# Внедрение кроссдокинга в логистической платформе Яндекс.Маркета

Уровень образования: бакалавриат

Направление *02.03.03 «Математическое обеспечение и администрирование информационных систем»*

Основная образовательная программа *СВ.5162.2021 «Технологии программирования»*

Научный руководитель:  
старший преподаватель, к.т.н. Литвинов Ю. В.

Рецензент:  
разработчик ПО, ООО «Яндекс.Технологии», Косолапов В. А.

Санкт-Петербург  
2025

Saint Petersburg State University

***Nikita Ksenchik***

Bachelor's Thesis

# Implementation of crossdocking in the Yandex.Market logistics platform

Education level: bachelor

Speciality *02.03.03 "Software and Administration of Information Systems"*

Programme *CB.5162.2021 "Programming Technologies"*

Scientific supervisor:  
Sc.D, senior lecturer Y.V. Litvinov

Reviewer:  
software developer "Yandex Technologies", V.A. Kosolapov

Saint Petersburg  
2025

# Оглавление

|                                                          |           |
|----------------------------------------------------------|-----------|
| <b>Введение</b>                                          | <b>4</b>  |
| <b>1. Постановка задачи</b>                              | <b>5</b>  |
| <b>2. Обзор</b>                                          | <b>6</b>  |
| 2.1. Обзор предметной области . . . . .                  | 6         |
| 2.2. Компоненты . . . . .                                | 7         |
| 2.3. Обзор существующей системы . . . . .                | 7         |
| 2.4. Обзор используемых технологий . . . . .             | 9         |
| <b>3. Архитектура</b>                                    | <b>10</b> |
| 3.1. Изменения в архитектуре . . . . .                   | 10        |
| <b>4. Описание решения</b>                               | <b>14</b> |
| 4.1. Изменения в бизнес-логике . . . . .                 | 14        |
| 4.2. Изменения в структуре хранения данных . . . . .     | 15        |
| 4.3. Изменения в архитектуре обработки событий . . . . . | 15        |
| <b>5. Развёртывание</b>                                  | <b>17</b> |
| 5.1. Поддержка на этапе развёртывания . . . . .          | 18        |
| <b>6. Внедрение</b>                                      | <b>19</b> |
| <b>Заключение</b>                                        | <b>21</b> |
| <b>Список литературы</b>                                 | <b>22</b> |

# Введение

В последние годы рынок электронной коммерции демонстрирует устойчивый рост, становясь неотъемлемой частью экономической жизни. Яндекс.Маркет как одна из ведущих платформ в этой сфере играет ключевую роль в развитии электронной торговли в России. Логистика на Яндекс.Маркете отвечает за эффективное управление заказами, поставками и возвратами, что напрямую влияет на удовлетворённость пользователей и общую эффективность работы платформы.

Существуют различные виды поставок, включая традиционные маршруты от поставщика через распределительный центр на склад. Однако логистика не ограничивается только этим путём – она включает в себя множество других маршрутов и процессов, направленных на обеспечение своевременной и качественной доставки товаров.

Несмотря на разнообразие логистических решений, количество маршрутов между распределительными центрами и складами остаётся ограниченным. Это создаёт определённые трудности в оптимизации логистических процессов, особенно в условиях растущего объёма заказов. Ограниченное количество маршрутов может приводить к задержкам в доставке, увеличению издержек и снижению общей эффективности работы платформы. Программное обеспечение играет ключевую роль в управлении и оптимизации этих процессов, позволяя более гибко и эффективно распределять грузы и ресурсы [12].

Возможным решением этих проблем может стать внедрение кроссдокинга [10] – процесса, при котором товары проходят через несколько распределительных центров перед тем, как попасть на нужный склад. Для реализации такого подхода необходимо разработать или адаптировать программное обеспечение, которое будет управлять потоками товаров и обеспечивать своевременное перемещение грузов между логистическими точками. Это позволит расширить логистическую сеть, увеличить количество маршрутов поставок и повысить общую эффективность логистических операций, что, в свою очередь, улучшит показатели работы платформы.

# 1. Постановка задачи

Целью работы является внедрение кроссдокинга в логистической платформе Яндекс.Маркета, основываясь на уже реализованных в программном обеспечении процессах для управления и оптимизации логистических процессов. Для её выполнения были поставлены следующие задачи:

1. разработать компонентную архитектуру программной поддержки процесса кроссдокинга;
2. реализовать программную поддержку процесса кроссдокинга в оркестраторе поставок логистической платформы;
3. произвести развёртывание программной поддержки процесса кроссдокинга в рабочей среде;
4. стабилизировать работу программной поддержки кроссдокинга.

## 2. Обзор

### 2.1. Обзор предметной области

Логистика Маркета включает три ключевых потока товаров: поток поставок, прямой поток и возвратный поток. В рамках потока поставок товары перемещаются со складов поставщика на склад Маркета. Прямой поток обеспечивает доставку продукции со складов Маркета конечному покупателю. Возвратный поток охватывает движение товаров, которые были возвращены клиентами или не дошли до них, обратно к поставщику.

Поставка начинается с того, что поставщик доставляет товары на распределительный центр (РЦ) Маркета. Здесь груз проходит этап приемки: работники проверяют его соответствие заявленному объему и качеству (нет явных повреждений), после чего выполняют подготовку к дальнейшему хранению и транспортировке. Товары (коробки с товарами) маркируются, упаковываются и формируются в паллеты — стандартные грузовые единицы, которые временно размещаются на складе РЦ.

Перед отгрузкой важно согласовать логистику: автомобиль прибывает в РЦ в заранее установленное время, чтобы минимизировать простой. Паллеты загружаются в транспорт с учетом конкретного направления — например, на целевой склад Маркета в другом городе. Водитель обеспечивает перевозку груза, а по прибытии работники склада принимают товар, проверяют его целостность и размещают в зонах длительного хранения.

Синхронизация данных между участниками цепочки играет критическую роль [11]. Распределительный центр и склады должны оперативно обмениваться информацией о времени прибытия грузов, их составе и статусах. Это позволяет избежать ошибок при планировании ресурсов и ускорить процессы приемки. Кроме того, поставщик получает возможность отслеживать движение товаров на каждом этапе — от приемки в РЦ до размещения на целевом складе. Такая прозрачность повышает

контроль над сроками поставок и повышает доверие поставщиков.

## 2.2. Компоненты

Ниже представлен список компонентов, которые будут обеспечивать программную поддержку процесса кроссдокинга.

SDS (Supply Delivery System) – высокоуровневый оркестратор, обеспечивающий мониторинг и управление статусами поставок на всех этапах.

GPMS (Goods Placement Management System) – компонент для управления грузоместами на этапе перевозки грузов между логистическими точками Маркета.

ПО РЦ – программное обеспечение распределительного центра.

ERP-система – система, используемая для бухгалтерского учёта и управления логистическими операциями.

ПИ – партнерский интерфейс. Интерфейс, через который поставщик взаимодействует с логистической платформой.

LDD (Logistics Data Dispatcher) – шина событий логистики маркета. Предоставляет возможность асинхронно обмениваться данными между одним или несколькими компонентами [4].

## 2.3. Обзор существующей системы

На рисунке 1 представлена упрощённая схема работы поставки.

### 1. Создание поставки

Поставка создаётся через партнерский или другой интерфейс. В SDS поступает заявка с указанием точек отправления и назначения. SDS фиксирует маршрут и ищет доступные слоты (время для отгрузки или приёмки) в нужные даты.

### 2. Создание приёмки на РЦ

SDS отправляет заявку на создание приёмки на РЦ. На этом этапе ещё неизвестно, какие именно товары привезёт поставщик. РЦ

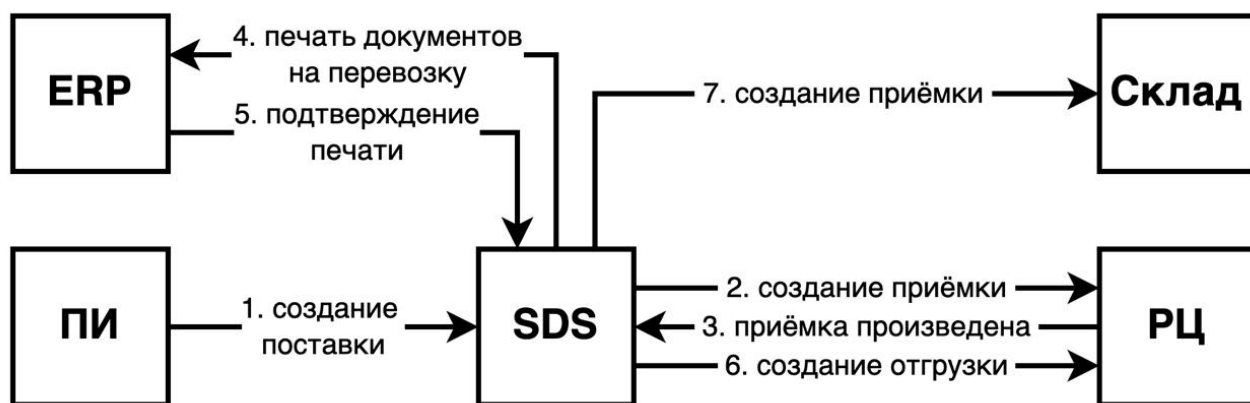


Рис. 1: Схема работы поставки до внедрения кроссдокинга

подтверждает возможность приёма товаров в указанный день.

### 3. Приёмка произведена

После завершения приёмки РЦ уведомляет SDS. SDS запрашивает реестр принятых товаров (коробок или паллет).

### 4. Создание отгрузки на РЦ и приёмки на складе

SDS создаёт отгрузку на РЦ и приёмку на складе, передавая реестр товаров, которые необходимо отгрузить и принять.

### 5. Создание документов

SDS запрашивает у ERP-системы печать документов, необходимых для транспортировки товаров.

### 6. Транспортировка

После получения документов начинается транспортировка груза. На каждом этапе поставки SDS обновляет статусы поставки и товаров, получая информацию от исполнителей. Это позволяет своевременно реагировать на аномалии в системе, утерю товаров, прогнозировать и пересчитывать даты операций, а также отображать необходимую информацию в партнёрском интерфейсе.

До внедрения кроссдокинга GPMS применялся только на прямом потоке товаров и в потоке поставок никак не участвовал.

## 2.4. Обзор используемых технологий

Java (11/17), Kotlin – основные языки программирования, используемые для реализации backend-сервисов.

MyBatis – фреймворк для объектно-реляционного отображения (ORM), упрощающий взаимодействие с базами данных [6].

Spring – фреймворк для построения масштабируемых и модульных Java-приложений.

PostgreSQL – реляционная система управления базами данных с открытым исходным кодом.

DBqueue – библиотека, предоставляющая реализацию очереди задач поверх Java и базы данных [9].

DBUnit – фреймворк, используемый для написания unit-тестов и интеграционных тестов. Предоставляет возможность описывать состояния базы данных в XML-формате [5].

YTesauros – платформа распределённого хранения и обработки больших данных с открытым исходным кодом. Используется для хранения логов различных компонентов Маркета.

YQL – декларативный язык запросов к системам хранения и обработки данных. Используется для анализа логов приложений.

## 3. Архитектура

Задача автора данного текста в рамках проекта – разработать архитектуру программной поддержки процесса кроссдокинга, взаимодействуя с коллегами из других команд логистики Маркета. Затем необходимо реализовать часть работы для компонента SDS, произвести развёртывание изменений и совместно с командой тестирования провести тестирование и стабилизацию системы.

### 3.1. Изменения в архитектуре

Новый тип маршрутов характеризуется наличием нескольких дополнительных плеч РЦ-РЦ. Для таких маршрутов необходимо создавать отгрузки и приёмы, отслеживать статусы грузомест и сверять их списки на каждом РЦ [7].

Добавление всей этой логики в SDS нежелательно, поскольку SDS задуман как верхнеуровневый оркестратор, который отдаёт команды более низкоуровневым сервисам, а сам концентрирует и актуализирует данные по поставке [3].

Поэтому было решено переиспользовать GPMS – сервис, который позволяет создавать, перевозить и отслеживать грузоместа на средней миле маршрута. Ранее GPMS применялся на прямом потоке товаров.

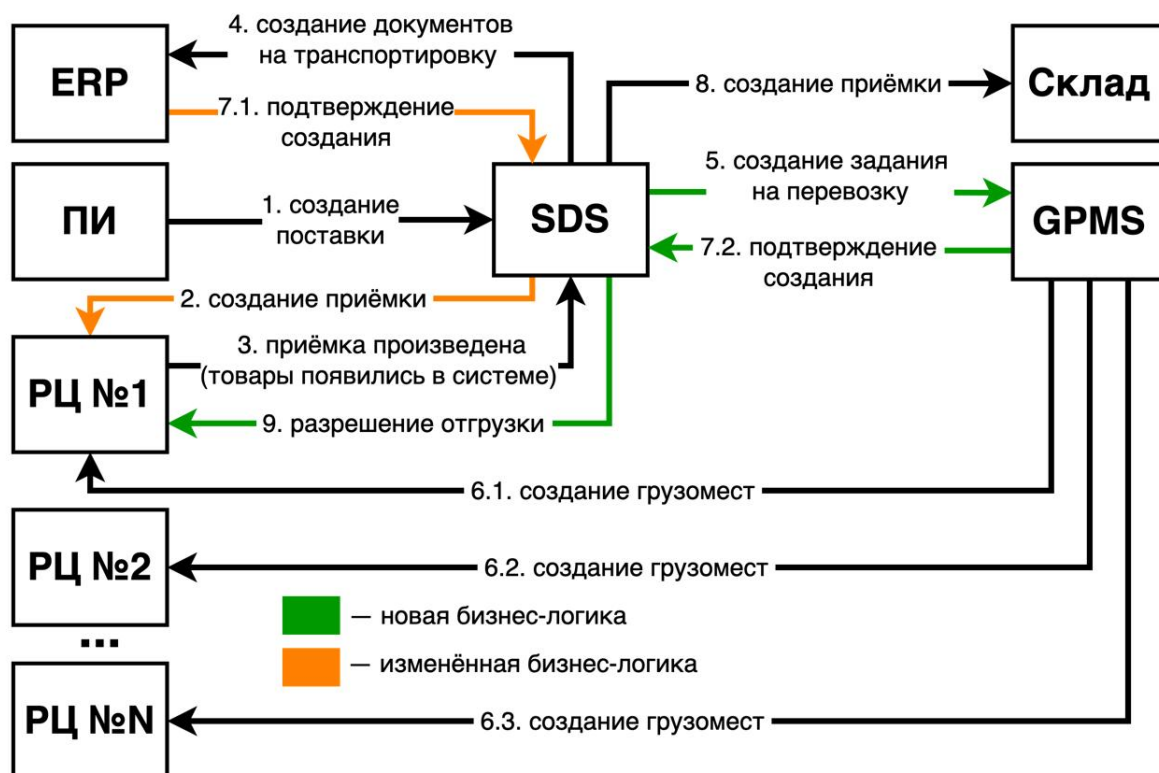


Рис. 2: Схема работы кроссдокинга

На рисунке 2 представлена схема работы кроссдокинга от момента создания поставки до момента отгрузки на РЦ1. Цветами выделена бизнес-логика, которая была добавлена или изменена автором данной работы.

### 1. Создание поставки

Первый шаг остался без изменений, поставщик через партнёрский интерфейс создаёт поставку.

### 2. Создание приемки на РЦ1

SDS ещё не знает, что конкретно будут принимать, поэтому создаёт заявку на приемку без планового реестра товаров. Также, так как в сложном маршруте несколько РЦ на средней миле, SDS передаёт информацию о следующей точке маршрута.

### 3. Приёмка произведена

На третьем шаге РЦ1 принимает поставку и передаёт SDS реестры с принятыми единицами товара.

4. Создание документов на транспортировку

5. Создание задания на перевозку

SDS, на основании имеющегося маршрута и полученного списка товаров, отправляет в GPMS задание на создание грузомест.

6. Создание грузомест

GPMS создаёт грузоместа на всех распределительных центрах, через которые поедет поставка (в том числе на первом).

7. Подтверждения от исполнителей

SDS дожидается подтверждения от ERP-системы о печати документов и подтверждений от GPMS о создании грузоместа на каждом из сегментов маршрута.

8. Создание приёмки на складе

SDS создаёт заявку на приёмку на конечном складе.

9. Разрешение отгрузки

Удостоверившись, что все необходимые задания созданы на всех необходимых исполнителях, SDS отправляет разрешение начала отгрузки на РЦ1.

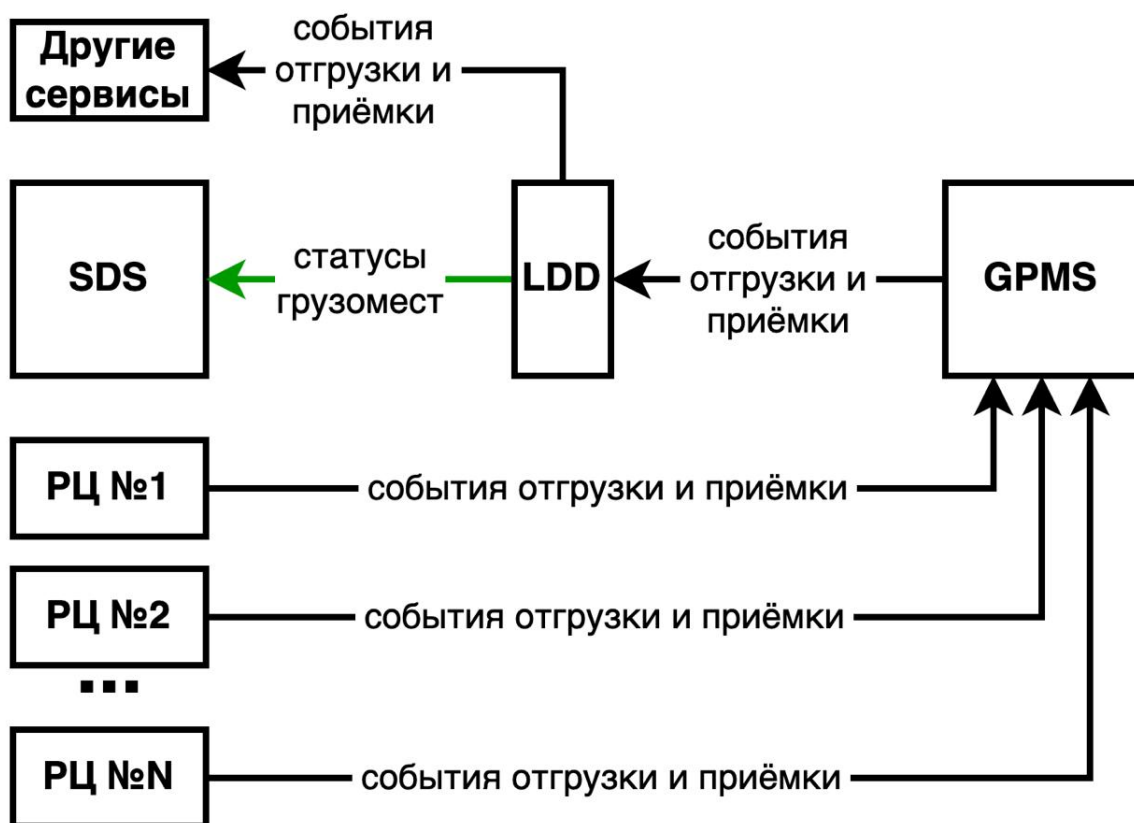


Рис. 3: Транспортировка на средней миле

На рисунке 3 представлена схема взаимодействия GPMS и SDS на средней миле. GPMS самостоятельно контролирует процесс перевозки грузомест между РЦ: создаёт отгрузки и приёмки, реагирует на аномальные ситуации. Статусы каждой единицы товара, принятой на РЦ1, GPMS фиксирует в специальной очереди в LDD. SDS читает события из этой очереди и обновляет статусы товаров, групп товаров, поставки и операций отгрузки и приёмки.

## 4. Описание решения

### 4.1. Изменения в бизнес-логике

В рамках доработки сервиса были реализованы следующие изменения в логике взаимодействия с внешними системами и построения маршрутов:

- Изменён формат передачи следующей логистической точки при создании задачи на приёмку товаров на распределительном центре (РЦ). Вместо фиксированных значений теперь осуществляется динамический поиск следующего пункта маршрута (РЦ или склад) на основе заранее сохранённого маршрута.
- Обновлена логика формирования и отправки подтверждения об отгрузке на РЦ1. Теперь подтверждение отправляется только после получения всех независимых подтверждений:
  - от ERP-системы – о создании сопроводительных документов;
  - от GPMS – о создании каждой карго-группы на каждой из соответствующих точек маршрута.

Для этого была реализована агрегация асинхронных событий с использованием DBqueue и соответствующих обработчиков [8].

- Изменена логика формирования запроса на создание грузомест в GPMS. Обновлён формат данных, а также механизм генерации сегментов маршрута — теперь он учитывает наличие нескольких точек маршрута.
- Реализована логика обработки событий приёма и отгрузки грузомест на средней миле. При поступлении события система определяет группу, к которой относится грузоместо, обновляет её статус и проверяет завершённость обработки всей группы. После обработки всех грузомест в рамках группы присваивается финальный статус.

- Все новые сценарии взаимодействия с GPMS покрыты интеграционными тестами.

## 4.2. Изменения в структуре хранения данных

- Для хранения и отслеживания состояния карго-групп добавлена новая таблица `request_cargo_place_group`. В таблице сохраняются ссылки на связанные задачи, идентификаторы логистических точек, текущие статусы и временные метки ключевых событий. Схема таблицы представлена на рисунке 4.

| <b>request_cargo_place_group</b> |                  |                    |
|----------------------------------|------------------|--------------------|
| <b>PK</b>                        | <b><u>id</u></b> | <b>BIGSERIAL</b>   |
| FK                               | request_id       | BIGINT NOT NULL    |
|                                  | cargo_group_id   | BIGINT NOT NULL    |
|                                  | status           | TEXT NOT NULL      |
|                                  | created_at       | TIMESTAMP NOT NULL |
|                                  | updated_at       | TIMESTAMP NOT NULL |

Рис. 4: Схема таблицы статусов карго-групп

- Для поддержки асинхронной обработки событий приёмки и отгрузки были добавлены очереди DBqueue. Каждое событие обрабатывается независимо, с фиксацией состояния в БД.

## 4.3. Изменения в архитектуре обработки событий

- В целях обработки подтверждений от систем ERP и GPMS была реализована координация событий в виде DBqueue-задачи: каждое входящее событие проверяет наличие второго, после чего при-

нимается решение о необходимости отправки итогового подтверждения отгрузки.

- Переработана логика генерации логистических операций на средней миле. Добавлены сервисы для создания задач на приёмку и отгрузку, с возможностью связывания их в рамках единой бизнес-операции.

## 5. Развёртывание

В процессе развёртывания новая логика работы через добавленные сервисы постепенно активировалась с помощью переключения feature-флагов [1]. При этом тщательно отслеживалась стабильность её функционирования.

### Этап 1

Цель этапа – перевести РЦ на новую систему формирования заданий для транспортировки грузов, заменив старую логику на работу с каргогруппами (группами грузомест).

Шаги этапа:

1. проверка корректности функционирования новой логики работы со всеми поставками по грузоместам (включая создание и отгрузку без отклонений);
2. перевод пилотных складов на новую логику работы с заданиями для транспортировки;
3. проверка, что новая логика работает корректно, переключение всех складов.

### Этап 2

Цель этапа – перевести РЦ на логику сборки палет. Активировать отправку события о консолидации, если следующая точка – РЦ.

### Этап 3

Цель этапа – настроить систему для построения маршрутов с несколькими точками.

Шаги этапа:

1. настроить доступные маршруты для конкретных поставщиков;

2. настроить логистический граф;
3. включить логику работы кроссдокинга.

## **5.1. Поддержка на этапе развёртывания**

После развёртывания изменений велось регулярное отслеживание мониторингов и логов. В случае возникновения ошибок проводился разбор причин с использованием логирования и внутренних инструментов поиска событий. При выявлении нестабильной работы отдельных компонентов производилось временное отключение соответствующих feature-флагов для обеспечения стабильной работы сервиса. Обнаруженные ошибки в коде устранялись по мере появления.

## 6. Внедрение

На этапе апробации система, уже подготовленная к осуществлению процесса кроссдокинга, тестировалась в физическом мире.

### Этап 1

Цель этапа – сделать тестовую поставку с помощью фиктивного поставщика и проверить успешность работы.

Тестовый поставщик, роль которого исполнял один из участников команды разработки, должен был поставить единицу товара по сложному маршруту на реальный склад.

Тестовая поставка проходила по следующим шагам:

1. оповещение дежурных всех участвующих в сложных маршрутах компонент, что будет создана тестовая поставка со сложным маршрутом в заданную дату;
2. регистрация тестового поставщика в системе и настройка ему возможности выбора начальной и конечной точки со сложным маршрутом;
3. создание поставки в партнёрском интерфейсе;
4. поставка условной единицы товара фиктивным поставщиком на склад;
5. отслеживание работы системы на всех этапах сложного маршрута.

### Этап 2

Цель этапа – произвести несколько поставок реальных поставщиков по сложному маршруту.

Этап проходил по следующим шагам:

1. выбор списка поставщиков, для которых можно настроить сложный маршрут;
2. настройка для них сложного маршрута;
3. отслеживание состояния системы в процессе.

### **Этап 3**

Цель этапа – запуск сложных маршрутов для всех поставщиков.

На данном этапе происходило постепенное включение возможности использования сложных маршрутов и отслеживание корректности работы системы с использованием канареечного развертывания [2].

# Заключение

В процессе работы были решены следующие задачи:

1. разработана компонентная архитектура программной поддержки процесса кроссдокинга;
2. реализована программная поддержка процесса кроссдокинга в оркестраторе поставок логистической платформы;
3. произведено развёртывание программной поддержки процесса кроссдокинга в рабочей среде;
4. работа программной поддержки кроссдокинга стабилизирована.

Код данной работы является собственностью компании ООО «Яндекс.Технологии» и находится под договором о неразглашении.

## Список литературы

- [1] Creator Web. Флаги функций (Feature Flags). — URL: [https://web-creator.ru/articles/feature\\_flags](https://web-creator.ru/articles/feature_flags) (дата обращения: 15 мая 2025 г.).
- [2] Fernandez Tomas. What Is Canary Deployment? — URL: <https://semaphore.io/blog/what-is-canary-deployment> (дата обращения: 15 мая 2025 г.).
- [3] Pati Ved Parkash. Event Driven Process Orchestration- A Practitioner's Viewpoint. — URL: <https://www.hcltech.com/blogs/event-driven-process-orchestration-a-practitioner-viewpoint> (дата обращения: 15 мая 2025 г.).
- [4] Services Amazon Web. Event buses in Amazon EventBridge. — URL: <https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-event-bus.html> (дата обращения: 15 мая 2025 г.).
- [5] Team DbUnit. DbUnit Official Documentation. — URL: <https://www.dbunit.org/> (дата обращения: 15 мая 2025 г.).
- [6] Team MyBatis. MyBatis 3 Documentation. — URL: <https://mybatis.org/mybatis-3/> (дата обращения: 15 мая 2025 г.).
- [7] Technologies GT. Кросс-докинг. Оптимизация логистики и потока товаров. — URL: <https://gt-tech.ru/stati/kross-doking.-optimizacziya-logistiki-i-potoka-tovarov> (дата обращения: 15 мая 2025 г.).
- [8] Tproger. Event-Driven архитектура: принципы и примеры использования. — URL: <https://tproger.ru/articles/event-driven-arhitektura--principy-i-primery-ispolzovaniya> (дата обращения: 15 мая 2025 г.).
- [9] YooMoney. db-queue: Queue processing library. — URL: <https://github.com/yoomoney/db-queue> (дата обращения: 15 мая 2025 г.).

- [10] Вышинская А. А. ТЕХНОЛОГИЯ “CROSS-DOCKING” В ЛОГИСТИКЕ. — URL: <https://rep.bntu.by/bitstream/handle/data/108647/181.pdf?isAllowed=y&sequence=1> (дата обращения: 9 апреля 2025 г.).
- [11] Иванов А. Н. Исследование информационных потоков в логистической системе. — URL: <https://cyberleninka.ru/article/n/issledovanie-informatsionnyh-potokov-v-logisticheskoy-sisteme> (дата обращения: 15 мая 2025 г.).
- [12] Кириллов А. В. Анализ и планирование развития логистической инфраструктуры маркетплейса с применением технологии больших данных (на примере компании ООО «Яндекс.Маркет»). — URL: <https://www.hse.ru/edu/vkr/466942349> (дата обращения: 9 апреля 2025 г.).