Технический проект

# Интернет-магазин “Мобайл”

Версия 01

[Интернет-магазин “Мобайл”](#_1fob9te)

[Глоссарий](#_olckanrswft6)

[1. Цель интеграции. Решаемая задача](#_gmwg35y6ra8n)

[2. Пользовательские требования](#_3cvz0usuhdmw)

[3. Требования к качеству](#_2yxiuss88oov)

[4. Общее описание интеграции систем](#_73kydfrhcc7r)

[4.1. Диаграмма потоков данных](#_44sinio)

[4.2. Регламент взаимодействия систем](#_gide5otph8rf)

[4.3. Общие требования к журналированию событий](#_xstjip3w86ie)

[4.4. Требования к обработке и протоколированию ошибок](#_k6l9da9w4hu2)

[4.5. Структура журнала интеграции](#_vqpu86ekdmvo)

[4.6. Применяемые технологии](#_j1ytxfyf01zv)

[4.7. Модель данных](#_oi8jc66uj6n9)

[5. Сценарии интеграции](#_83wsu1trspky)

[5.1. Интеграция Сервис “OnLite”- Сервис УЦ](#_whgq5cuineh2)

[5.2. Диаграмма последовательности](#_ss957s579qtb)

[6. Передача и преобразование данных](#_3j2qqm3)

[Описание данных, передаваемых из OnLite в УЦ](#_ou94od53kop)

[Описание данных, передаваемых из Мобильного приложения в Service Desk](#_89anzenymvjn)

[Описание данных, передаваемых из системы ЕСИА в систему OnLite](#_gu9nl66fottm)

[7. Описание технологии передачи данных](#_2gyc3zttsy5z)

[7.1. REST API](#_veghg9dcl8t3)

[7.1.1. Запрос на выпуск ЭП в УЦ](#_s3f7sk1z7epj)

[7.1.2. Отправка обращения пользователя](#_2vf7zh2x5qop)

[7.1.3. Метод получения данных пользователя из ЕСИА](#_5vykjht8b7ny)

[7.2. SOAP](#_h2kciv75fjbe)

[7.2.1. Краткое описание метода](#_fe4ygcp5i6vi)

[7.2.2. Примеры запросов и ответов](#_xozi4s5p10iu)

## Глоссарий

|  |  |
| --- | --- |
| **Термин, Сокращение** | **Описание** |
| Мобайл (сайт) | Веб-сайт, через которое клиент осуществляет покупку |
| СОЗ | Служба обработки заказов. Система, которая принимает заказы с сайта и проводит дальнейшую обработку. Является оркестратором процесса. |
| БС | Биллинговая система. Система, которая учитывает цену товара в зависимости от количество товаров, скидки по товарам, налоги и выставляет счет для оплаты |
| БЭ | Платежная система. Система, через которую принимается платеж от клиента. В текущем случае это банк эквайринг |
| СП(С) | Система продуктов. Каталог товаров, доступный для выбора на сайте Клиенту |
| СД | Служба доставки СДЭК. Система, реализующая доставку товаров клиенту по указанному в заказе адресу |

## Цель интеграции. Решаемая задача

Цель: увеличение прибыли компании по продаже гаджетов, реализация продуктов компании.

Задача: спроектировать процесс реализации товаров через интернет-магазин.

Реализовать интеграцию со следующими сервисами:

1. служба обработки заказов (СОЗ)+сайт СОЗ(С)
2. служба доставки (СДЭК) СД
3. платежная система, (банк эквайринг) БЭ
4. биллинговая система, (количество товаров во взаимосвязи со стоимостью и скидки по товарам, налоги) - сколько выбрал, по какой цене, когда решил оплатить идет в платежную систему БС
5. система продуктов (товаров)+склад СП(С)

## Пользовательские требования

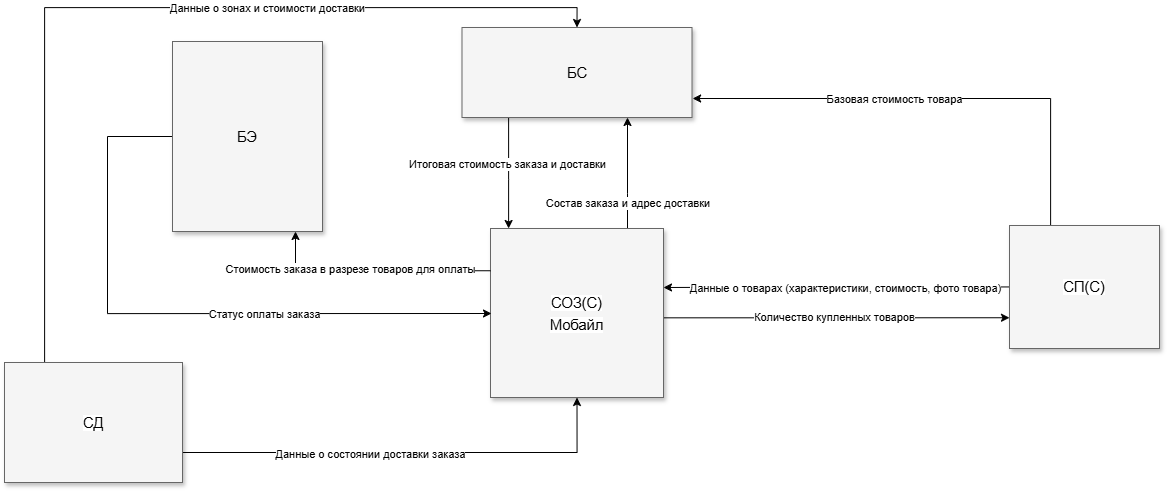
|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Код требования** | **Требование** |
| 1 | СОЗ(с). Создание заказа | Создает заказ, полученный на сайте от клиента. Фиксирует данные: заказчик, состав заказа (список продуктов + количество), адрес доставки. |
| 2 | БС. Расчет стоимости заказа. | Получает информацию о заказе из СОЗ(С) и рассчитывает стоимость товара, заказа, доставки согласно информации СП |
| 3. | СОЗ(с). Отслеживание заказа. | Сотрудник может посмотреть статус обработки заказа |
| 4. | Соз(С). Уведомление клиента | Клиенту отправляется уведомление о статусе и состоянии заказа на выбранный канал связи |
| 5. | СД. Уведомление клиента | Клиенту отправляется уведомление о статусе и состоянии посылки на выбранный канал связи |
| 6. | БЭ.Проведение оплаты | Проведение и подтверждение факта оплаты клиентом, клиенту отправляется чек |
| 7. | СП(С). Проверка наличия товара | При расчете заказа учитывается наличие товара на складе, учитывается стоимость |
| 8. | СОЗ(с).передача информации в СДЭК для доставки заказа | Передача информации об оплаченном заказе в СДЭК |
| 9. | БС.применение промоакции | При расчете стоимости заказа учитываются промоакции (скидки, в том числе от объема заказанного товара) |
| 10. | СД. Отслеживание посылки | Клиент может посмотреть статус доставки посылки |

## Требования к качеству

|  |  |
| --- | --- |
| **Код требования** | **Требование** |
| **Производительность** | |
| Производительность. Число пользователей | В период пиковой нагрузки не более 100 000 пользователей. |
| Производительность. Создание заказа | Создание заказа в системе не более 1 минуты:  а) Получение данных о товаре (из СП)  б) Расчет итоговой стоимости заказа (из БЛ)  в) Расчет доставки (из СД) |
| Производительность. Проведение платежа | Проведение платежей не более 3 минут |
| Производительность. Отправка уведомлений | Отправка уведомления не более 3 минут от момента смены статуса |
| **Надежность** | |
| Надежность. Вероятность сбоя | 2% |
| Надежность. Время восстановления после сбоя | не более 5 мин |
| **Доступность** | |
| Доступность. Допустимое время простоя в час, минут | 98%  1 мин |
| Доступность.  Допустимое время простоя в месяц, часов | 1 час |
| Время предоставления сервиса | 24\*7 |
| Архивирование информации | Ежечасно |
| **Безопасность** | |
| Безопасность. Контроль целостности данных в БД | При сбоях проведения транзакции обеспечить откат без потери данных. |
| Безопасность. Антибот | Ограничение количества запросов с одного IP адреса к серверу |

## Общее описание интеграции систем

### Диаграмма потоков данных



### Регламент взаимодействия систем

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Система-источник** | **Система-приемник** | **Передаваемые данные** | **Полный объем/ изменения** | **Периодичность обмена/ событие** |
| 1 | СП(С) | СОЗ(с) | Данные о товарах и ценах | Полный объем | По запросу |
| 2 | СП(С) | СОЗ(с) | Данные о товарах и ценах | Изменения | По событию (изменения в СП) |
| 3 | СОЗ(с) | БС | Состав заказа | Полный объем | По запросу |
| 4 | СП(с) | БС | Базовая стоимость товара | Полный объем | По запросу |
| 5 | СД | БС | Данные о зонах и стоимости доставки | Полный объем/изменения | По событию (получен заказ на расчет стоимости или по расписанию API сдэк ) |
| 6 | БС | СОЗ(с) | Итоговая стоимость заказа | Полный объем | По событию (завершил расчет) |
| 7 | СОЗ(с) | БЭ | Стоимость заказа для оплаты | Полный объем | По запросу |
| 8 | БЭ | СОЗ(с) | Статус оплаты заказа | Полный объем | По событию (оплачено\отказ в оплате) |
| 9 | Соз(с) | СД | Данные о заказе | Полный объем | По событию (оплачен заказ) |
| 10 | СД | Соз(с) | Данные о состоянии доставки | Изменения(приехал/в пути) | По событию (заказ передан клиенту) |
| 11 | СОЗ(с) | СП(с) | Данные о товарах | Изменения | По событию (оплата заказа)  \* для изменения количества товаров |

### Общие требования к журналированию событий

* Конечный результат передпч данных

|  |  |  |
| --- | --- | --- |
| **Код требования** | **Требование** | **Система** |
| Журналирование. Сохранение записей о получении данных о товарах и ценах | В рамках журналирования должна сохраняться информация о событиях получения данных о товарах и ценах | СОЗ(С) |
| Журналирование. Сохранение записи о получении данных о зонах и стоимости доставки | В рамках журналирования должна сохраняться информация о событиях получения данных о зонах и стоимости доставки | БС (с) |
| Журналирование. Сохранение записей о получении итоговой стоимости заказа | В рамках журналирования должна сохраняться информация о событиях получения итоговой стоимости заказ | СОЗ (С) |
| Журналирование. Сохранении записи о получении информации об оплате заказа | В рамках журналирования должна сохраняться информация о событиях получения информации о статусе оплаты заказа | СОЗ (С) |
| Журналирование. Сохранение записи о событиях изменения статуса доставки заказа | В рамках журналирования должна сохраняться информация о событиях получения информации о статусе доставки заказа | СОЗ (С) |
| Журналирование. Сохранение записи о событиях получения данных об изменении количества товара на складе | В рамках журналирования должна сохраняться информация о событиях получения информации об изменении количества товаров на складе | СП (С) |
| Журналирование. Сохранение записи о результате взаимодействия систем | В рамках журналирования должна автоматически сохраняться информация по событиям передачи данных, которые регистрируются и обрабатываются в Системе.  Информация должна сохраняться в виде записи в журнале.  Запись в журнале должна содержать следующие блоки с данными:   * источник события * приемник события * результат события * сколько времени заняла передача * дата и время передачи данных |  |
| Журналирование. Формат записи в журнале | Запись в журнале должна содержать следующие блоки с данными:   * источник события: информационная система * операция передачи или получения данных, выполняемая в Системе, * вызываемые сторонние (внешние) системы, * результат передачи или получения данных, * ссылка на идентификатор данных в информационной системе |  |
| Журналирование. Отчет | Система должна предоставлять возможность на основе результата выполнения запроса к журналу формировать отчет для дальнейшего разбора информации.  Отчет может быть реализован в виде лог-файла или excel |  |
| Журналирование. Отчет.Режимы формирования | Подсистема журналирования должна предоставлять возможность формировать отчет в режиме:   * Автоматическое формирование отчета согласно правилам формирования с указанием частоты формирования (ежедневно). * Автоматическое формирование отчета на данный момент времени в интерфейсе администратора в соответствии с форматом |  |
| Журналирование. Период хранения записей | Подсистема журналирования должна предоставлять возможность хранения записей в течение 6 месяцев, после чего все устаревшие записи подлежат удалению. |  |

### Требования к обработке и протоколированию ошибок

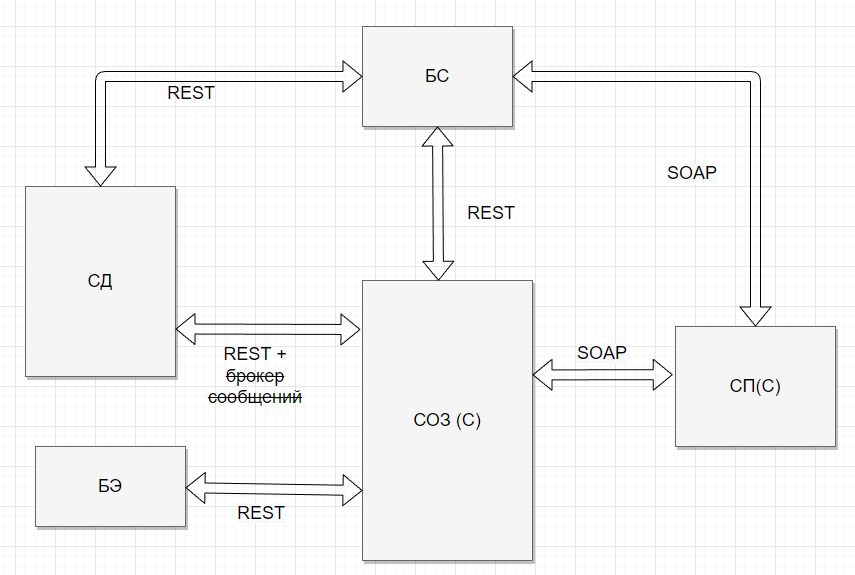
|  |  |
| --- | --- |
| **Код требования** | **Требование** |
| Ошибка авторизации | Система должна фиксировать факт не успешной авторизации с кодами ошибок и текстовыми сообщениями |
| Ошибка исполнения сервиса получения данных | Система должна фиксировать следующие ошибки при вызове методов получения данных:   * success - успешное выполнение метода * timeout - ошибка исполнения по истечению времени выполнения метода * failed - ошибка выполнения бизнес-правила * error - ошибка исполнения по несоответствию справочных данных и\или формата сообщения. |
| Ошибка на стороне внешней системы | Система должна фиксировать все ошибки получения ответов с кодами ошибок и текстовыми сообщениями |
| Ошибка на стороне вызывающей системы | Система должна фиксировать все ошибки отправки запросов с параметрами запросов |
| Уведомление администратору об ошибках исполнения в статусе error | Система иметь возможность отображать сообщения о текущих ошибках исполнения сервиса получения данных в статусе error администратору системы в режиме реального времени   * Уведомление в логе журнала интеграции с временем ошибки * Уведомление по e-mail |
| Пользовательское уведомление об ошибках бизнес-правил | Система должна отображать сообщение об ошибке с типом failed |
| Пользовательское уведомление об ошибке получения данных | Система должна отображать сообщение об ошибке обновления данных из внешних систем клиенту в формате “Ошибка получения данных, обратитесь в службу поддержки” |

### Структура журнала интеграции

Новая запись добавляется в журнал при каждом вызове методов обращения к **Service Desk, УЦ, ЕСИА, АБС**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Описание** | **Заполнение** |
| LogId | ИД записи в журнале | Автоматически |
| SysId | ИД внешней системы | Из справочника |
| attributes | Атрибуты (ID) | По шаблону |
| Status | Статус | 1-success  2-timeout  3-failed  4 -error |
| RequestTime | Время вызова сервиса | Заполняется в момент вызова сервиса |
| ResponceTime | Время отработки запроса | Заполняется в момент возврата ответа |

### Применяемые технологии



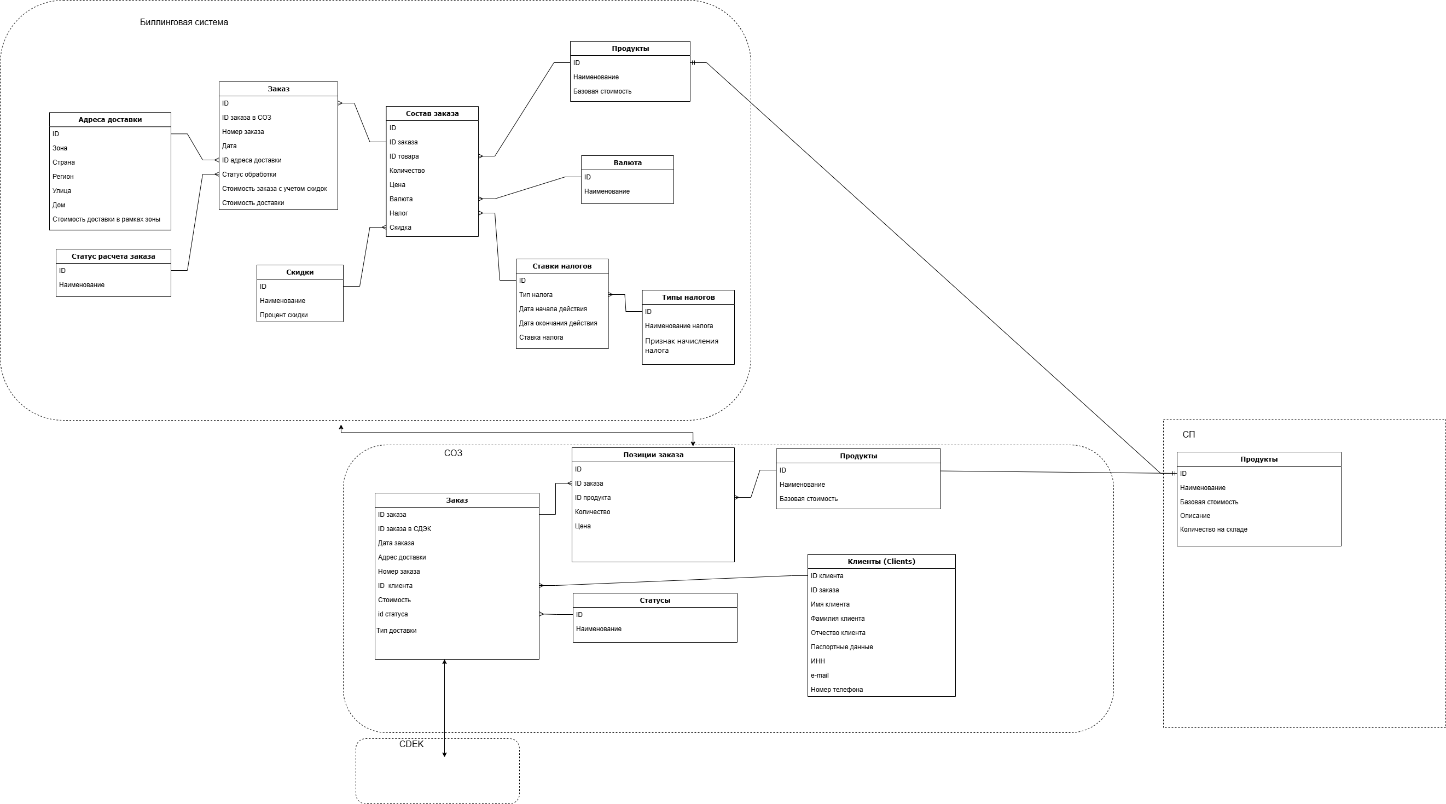
Технологии:

* HTTP/REST;
* HTTP/SOAP;
* Передача файлов.

Инструменты тестирования и документирования решения:

* PostMan
* SoapUI
* Swagger

### Модель данных



## Сценарии интеграции

### Интеграции

БД - база данных

БС - биллинговая система

БЭ - платежная система, через которую принимается платеж от клиента (банк эквайринг)

СОЗ(С) - служба обработки заказов, сайт “Мобайл”

СП(С) - система продуктов, склад  
СД - служба доставки СДЭК ИС СДЭК

ИС - Информационная система.

|  |
| --- |
| **ID:** UC-4 |
| **Название:** Получение базовой стоимости товара |
| **Участники:** БС, СП(С) |
| **Предусловия:** В БС получен заказ на расчет стоимости**Триггер:** БС отправляет запрос в СП(С)**Постусловия**: БС получает информацию о базовой стоимости товара |
| **Основной поток:**   1. БС инициирует запрос и передает данные о товаре в заказе с СП(С):    1. идентификатор каждого товара в заказе. 2. СП(С) обрабатывает запрос, получает данные. 3. СП(С) по каждому идентификатору находит товар и его характеристики (базовая стоимость). 4. СП(С) возвращает в БС:    1. идентификатор каждого товара в заказе с базовой стоимостью (без учета скидок). 5. БС сохраняет полученные данные для дальнейшего расчета скидок и итоговой стоимости заказа. 6. Сценарий завершен |
| **Расширение:** 3а. По полученному идентификатору СП(С) не находит товар.  3а.1. СП(С) возвращает ошибку в БС об отсутствии товара по указанному идентификатору  3а.2. Сценарий завершен  3б. По найденному товару не указана базовая стоимость  3б.1. СП(С) возвращает ошибку в БС об отсутствии стоимости по указанному идентификатору  3б.2. Сценарий завершен |

|  |
| --- |
| **ID:** UC-5 |
| **Название:** Получение данных о зонах и стоимости доставки |
| **Участники:** БС, СД |
| **Предусловия**: В БС получен заказ на расчет стоимости**Триггер:** БС отправляет запрос на стоимость доставки в СД**Постусловия**: В БС получена стоимость доставки в соответствии с параметрами заказа |
| **Основной поток:**   1. БС передает данные о заказе с СД:    1. идентификатор заказа    2. способ доставки (в пункт СДЭК, курьер, экспресс-доставка)    3. адрес доставки 2. СД получает данные. 3. СД в соответствии с полученными данными возвращает ответ:    1. идентификатор заказа    2. стоимость доставки 4. БС сохраняет полученные данные для дальнейшего расчета итоговой стоимости заказа. 5. Сценарий завершен. |
| **Расширение:** 1а. Не получен один из параметров запроса.  1а.1. СД возвращает ошибку в БС о недостаточности данных для расчета  1а.2. Сценарий завершен  1б. Указан адрес, по которому не осуществляется доставка  1б.1. СД возвращает ошибку с предложением выбрать другой адрес или способ доставки  1б.2. Сценарий завершен |

|  |
| --- |
| **ID:** UC-3 |
| **Название: Передача данных заказа в биллинговую систему** |
| **Участники:** СОЗ(С), БС, пользователь |
| **Предусловия**:  1. Пользователь в СОЗ(С) выбрал необходимый продукт (товар) и указал его количество 2. Пользователь в СОЗ(С) указал адрес доставки 3. В СОЗ(С) создан заказ  **Триггер:** Пользователь в СОЗ(С) нажал кнопку “Перейти к расчету и оформлению заказа”**Постусловия**:  1. В БС поступили данные заказа для расчета стоимости 2. БС вернул ответ об успешном вызове метода |
| **Основной поток:** Пользователь в СОЗ(С) инициирует запрос путем нажатия кнопки “Перейти к расчету и оформлению заказа”  1. СОЗ(С) инициирует запрос в БС для передачи данных заказа, со следующими атрибутами:  * идентификатор заказа в БД СОЗ(С) * идентификатор продукта (товара) в заказе * количество выбранного продукта (товара) в заказе * дата/время * адрес доставки заказа  1. БС успешно обрабатывает запрос - сохраняет данные заказа в БД, устанавливает статус “В обработке” 2. БС возвращает в СОЗ(С) ответ об успешном вызове метода (201 Created) + идентификатор сохраненного заказа в БД БС 3. СОЗ(С) сохраняет и связывает идентификатор заказа, присвоенный в БД БС, с идентификатором заказа в БД СОЗ(С) 4. Сценарий завершен |
| **Расширение:** **2а. Пользователь не указал адрес доставки заказа.**  2а1. СОЗ(С) выводит сообщение об отсутствии адреса доставки для осуществления полного расчета стоимости заказа с учетом доставки.  2а2. Сценарий завершен.  **3а. Методы API БС недоступны/нет ответа (таймаут).**  3а1. СОЗ(С) не получает ответа от БС в течении определенного времени (таймаута)/ БС возвращает код ответа типа 5\*\*  3а2. СОЗ(С) осуществляет повторную попытку отправки запроса  3а3. Если повторные попытки в течении определенного времени не были успешными, то СОЗ(С) выводит сообщение пользователю об ошибке.  3а2. СОЗ(С) создает запись в журнале логов об ошибке.  3а3. Сценарий завершен. |

|  |
| --- |
| **ID:** UC-6 |
| **Название: Передача итоговой стоимости заказа по факту выполнения расчета** |
| **Участники:** СОЗ(С), БС |
| **Предусловия**:Из СД успешно получена стоимость доставки в соответствии с параметрами заказа (UC-5)Из СП(С) успешно получена базовая стоимость товаров (без учета скидок) (UC-4)  1. БС произвел расчет стоимости доставки и продуктов (товаров) в заказе с учетом промо акций/скидок  **Триггер:** БС успешно завершил расчет стоимости доставки и заказа. Статус заказа в БД БС установлен на “Расчет завершен”**Постусловия**: В СОЗ(С) получена и сохранена итоговая стоимость заказа и доставки |
| **Основной поток:** БС передает СОЗ(С) данные об итоговой стоимости заказа и доставки:  * идентификатор заказа в БД БС (для этапа валидации) * идентификатор заказа в БД СОЗ(С) * идентификатор продукта (товара) в заказе * стоимость продукта (товара) с учетом скидок/акций * валюта продукта (товара) * количество выбранного продукта (товара) в заказе * дата/время * адрес доставки заказа * стоимость доставки заказа * общая сумма ИТОГО  СОЗ(С) получает данные и проводит валидацию: по полученным идентификаторам определяет заказ, для которого пришел расчет стоимостиСОЗ(С) сохраняет стоимость заказа и доставкиСОЗ(С) отправляет ответ БС об успешном получении данныхСценарий завершен |
| **Расширение:** **1а. При расчете стоимости заказа произошла ошибка (СП(С) вернула ошибку в БС об отсутствии товара по указанному идентификатору/ или для указанного адреса доставки СД вернула ошибку в БС)**  1а1.БС передает СОЗ(С)   * идентификатор заказа в БД БС * идентификатор заказа в БД СОЗ(С) * код ошибки   1а2. Сценарий завершен.    **2а. СОЗ(С) определяет, что заказа, для которого передана итоговая стоимость, не существует**  2а1. СОЗ(С) сообщает БС об ошибке:   * идентификатор заказа в БД БС * код ошибки   2а2. Сценарий завершен.    **2b. Методы API СОЗ(С) недоступны.**  2b1. СОЗ(С) возвращает код ответа типа 5\*\*  2b2. БС выполняет повторную попытку вызова пока метод не вернет код отличный от 5\*\* (Гарантированная доставка) |

|  |
| --- |
| **ID:** UC-11 |
| **Название: Передача данных о купленных продуктах(товарах) для учета на складе** |
| **Участники:** СОЗ(С), СП(С) |
| **Предусловия**: Оплата заказанных продуктов(товаров) прошла успешно**Триггер:** БЭ (платежная система) вернула в СОЗ(С) данные о платеже со статусом “Проведен”**Постусловия**: В СП(С) получены идентификаторы продуктов (товаров), которые оплачены в заказе |
| **Основной поток:**   1. СОЗ(С) передает данные в СП(С) о купленных продуктах(товарах) для списания на складе (изменения признака наличия/количества):  * идентификатор оплаченного заказа в БД СОЗ(С) * идентификатор продукта (товара) оплаченного в заказе * количество продукта (товара) оплаченного в заказе * дата/время  1. СП(С) по полученным идентификаторам определяет продукты (товары) 2. СП(С) вносит изменения в БД по наличию/количеству товара 3. СП(С) отправляет ответ об успешном получении данных 4. Сценарий завершен |
| **Расширение:**   **2а. СП(С) определяет, что товар, по которому передан идентификатор был ранее списан/продан (у товара установлен уже статус “Нет в наличии”)**  2а1. СП(С) сообщает СОЗ(С) об ошибке:   * идентификатор продукта (товара) оплаченного в заказе * код ошибки   2а2. Сценарий завершен.    **2b. Методы API СП(С) недоступны.**  2b1. СП(С) возвращает код ответа типа 5\*\*  2b2. СОЗ(С) выполняет повторную попытку вызова пока метод не вернет код отличный от 5\*\* (Гарантированная доставка) |

## Интеграция СОЗ-БЭ

# Сценарий

\*БК-Банковская карта

**Важно!** Процесс интернет-эквайринга - это сложный процесс взаимодействия нескольких систем: БЭ, процессингового центра, банка-эмитента и онлайн кассы. Поэтому для упрощения примем, что БЭ содержит в себе функции эквайринга и процессинга, И данные БК вводятся на странице СОЗ (а не на странице БЭ). Процесс фискализации не рассматривается.

**Границы процесса**

В данном UC я рассматриваю весь процесс оплаты заказа с однофакторной аутентификацией, сформированного в СОЗ, который начинается, когда Клиент нажимает “Оплатить” на странице СОЗ, и заканчивается, когда СОЗ получила от БЭ позитивный статус оплаты или позитивный url.

Данный процесс является искусственным и в жизни он вряд ли существует.

|  |
| --- |
| **UC\_Х.Х\_Оплата заказа, сформированного в СОЗ** |
| **Участники**: Система, БЭ |
| **Предусловия**: Клиент на сайте СОЗ перешел на страницу оплаты заказа и ввел данные БК. |
| **Триггер**: Клиент нажал кнопку “Оплатить”. |
| **Основной сценарий:**   1. Система формирует запрос **SrvName\_1\_Rq** (описание формата и правил формирования запроса см. в разделе \*ссылка на раздел документа\*) и отправляет его в БЭ. 2. БЭ принимает запрос от Системы и проверяет его на обязательность заполнения данных. Если обязательные параметры не переданы И/ИЛИ формат данных не соответствует форматам, зафиксированных на этапе проектирования, то см. AUC\_X.X\_1. 3. Если обязательные поля получены, БЭ проводит следующие проверку реквизитов БК в подсистеме процессинга:    1. зарегистрирован ли в подсистеме процессинга номер БК, переданный в **param\_1** запроса **SrvNameRq,** со сроком действия, переданным в **param\_2** запроса **SrvNameRq;**    2. является ли текущая календарная дата поздней по отношению к сроку действия БК, переданного в **param\_2** запроса **SrvNameRq**;    3. является ли код CVV, переданный в параметре **param\_3** запроса **SrvNameRq**, валидным для номера карты, переданного в **param\_1** запроса **SrvNameRq;**    4. доступна ли для списания с карты сумма, значение которой передано в параметре **param\_3** запроса **SrvNameRq**. 4. Если хотя бы одна проверка не пройдена, см. AUC\_X.X\_2. 5. Если все проверки пройдены, то БЭ формирует ответ **SrvName\_1\_Rs** (описание формата и правил формирования ответа см. в разделе \*ссылка на раздел документа\*), параметр **param\_4** с кодом ответа заполняется значением “0” (см. описание кодов ответов в разделе \*ссылка на раздел документа\*). |
| **Постусловия**:   1. Старт сценария в БЭ **UC\_Х.Х\_Формирование запроса в банк-эмитент на списание денежных средств** (\*ссылка на раздел документа\*). 2. Старт сценария в БЭ **UC\_Х.Х\_Формирование запроса в онлайн-кассу для фискализации** (\*ссылка на раздел документа\*). 3. СОЗ визуализирует Клиенту сообщение с текстом: “Оплата проведена успешно” |

|  |
| --- |
| **AUC\_X.X\_1\_Недостаточность данных в запросе** |
| **Участники**: Система, БЭ |
| **Предусловия**: Клиент на сайте Интернет-магазина перешел на страницу БЭ для оплаты и ввел данные БК. |
| **Триггер**: Клиент нажал кнопку “Оплатить”. |
| **Основной сценарий:**   1. Система формирует запрос **SrvName\_1\_Rq** (описание формата и правил формирования запроса см. в разделе \*ссылка на раздел документа\*) и отправляет его в БЭ. 2. БЭ принимает запрос от Системы и проверяет его на обязательность заполнения данных. Если обязательные параметры не переданы И/ИЛИ формат данных не соответствует форматам, зафиксированных на этапе проектирования, то БЭ формирует ответ **SrvName\_1\_Rs** (описание формата и правил формирования ответа см. в разделе \*ссылка на раздел документа\*), где параметр **param\_4** заполняется соответствующим кодом ошибки (см. описание кодов ответов в разделе \*ссылка на раздел документа\*). |
| **Постусловия**:   1. СОЗ визуализирует Клиенту сообщение с текстом: “Извините, оплата не проведена. Повторите попытку позднее” |

|  |
| --- |
| **AUC\_X.X\_2\_БК не прошла проверки** |
| **Участники**: Система, БЭ |
| **Предусловия**: Клиент на сайте Интернет-магазина перешел на страницу БЭ для оплаты и ввел данные БК. |
| **Триггер**: Клиент нажал кнопку “Оплатить”. |
| **Основной сценарий:**   1. Система формирует запрос **SrvName\_1\_Rq** (описание формата и правил формирования запроса см. в разделе \*ссылка на раздел документа\*) и отправляет его в БЭ. 2. БЭ принимает запрос от Системы и проверяет его на обязательность заполнения данных. Если обязательные параметры не переданы И/ИЛИ формат данных не соответствует форматам, зафиксированных на этапе проектирования, то см. AUC\_X.X\_1. 3. Если обязательные поля получены, БЭ проводит следующие проверку реквизитов БК в подсистеме процессинга:    1. зарегистрирован ли в подсистеме процессинга номер БК, переданный в **param\_1** запроса **SrvNameRq,** со сроком действия, переданным в **param\_2** запроса **SrvNameRq;**    2. является текущая календарная дата поздней по отношению к сроку действия БК, переданного в **param\_2** запроса **SrvNameRq**;    3. является ли код CVV, переданный в параметре **param\_3** запроса **SrvNameRq**, валидным для номера карты, переданного в **param\_1** запроса **SrvNameRq;**    4. доступна ли для списания с карты сумма, значение которой передано в параметре **param\_3** запроса **SrvNameRq** 4. Если хотя бы одна проверка не пройдена, то БЭ формирует ответ **SrvName\_1\_Rs** (описание формата и правил формирования ответа см. в разделе \*ссылка на раздел документа\*), где параметр **param\_4** заполняется соответствующим кодом ошибки (см. описание кодов ответов в разделе \*ссылка на раздел документа\*). |
| **Постусловия**:   1. СОЗ визуализирует Клиенту сообщение с текстом:    1. “Недостаточно средств на карте”, если значение параметра **param\_4** ответа **SrvName\_1\_Rs** равно “n”;    2. “Истек срок действия карты”, если значение параметра **param\_4** ответа **SrvName\_1\_Rs** равно “m”;    3. “Карта не авторизована”, если значение параметра **param\_4** ответа **SrvName\_1\_Rs** равно “k”. |

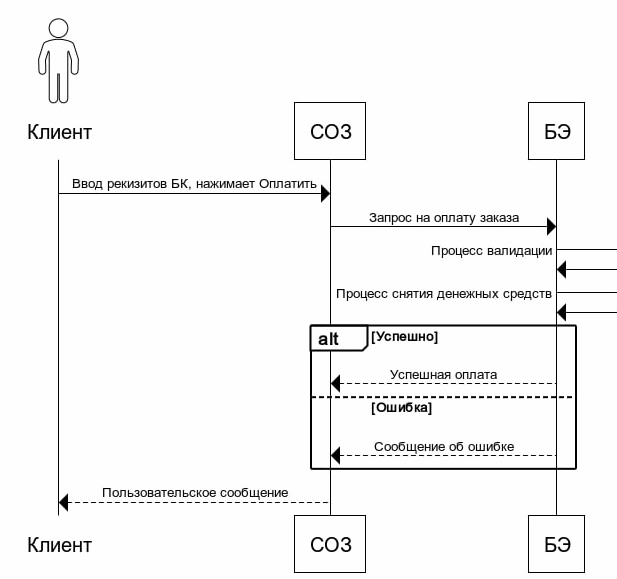
## Интеграция СОЗ- СД

|  |
| --- |
| **ID:** UC-9 |
| **Название: Регистрация заказа в службе доставки СДЭК** |
| **Участники: СОЗ, СД, Пользователь СОЗ** |
| **Предусловия**:  1. В СОЗ создан и оплачен заказ 2. Настроено подключение к сервису СДЭК  **Триггер:** В СОЗ статус заказа изменился на “Оплачен” **Постусловия**:  1. Заказ зарегистрирован в ИС СДЭК. 2. В СОЗ Статус заказа изменен на “Принят к исполнению” |
| **Основной поток:**   1. СОЗ выполняет авторизацию в сервисе СД. 2. Авторизация пройдена успешно. Сервис СД возвращает токен. 3. СОЗ инициирует запрос в CД на регистрацию заказа со следующими атрибутами:    * Тип заказа    * Номер заказа    * Дата заказа    * Адрес доставки    * Информация о товаре    * Данные получателя 4. СД получает и обрабатывает запрос: инициирует регистрацию заказа в ИС СДЭК. 5. Заказ зарегистрирован. СД отправляет в СОЗ ответ на регистрацию заказа:    * идентификатор заказа в ИС СДЭК,    * статус запроса - “Обработан”. 6. СОЗ сохраняет идентификатор заказа в ИС СДЭК. 7. СОЗ переводит заказ в статус “Принят к исполнению”. 8. Сценарий завершен. |
| **Расширение:****2а. Авторизация не пройдена.**2а1. Формируется повторный запрос авторизации.2а2. В случае неуспешного сценария, ошибка фиксируется в логе.2а3. Сценарий завершен **5а. Заказ не зарегистрирован.**  5а1. СД отправляет в СОЗ ответ на регистрацию заказа:   * статус запроса- “Обработан с ошибкой” * ошибки, возникшие в ходе выполнения запроса.   5а2. СОЗ сохраняет результат обработки запроса в карточку запроса.  5а3. СОЗ переводит заказ в статус “Ошибка передачи на исполнение”.  5а4. Пользователь СОЗ устраняет ошибки в оформленном заказе. Переводит в заказ статус “Повторная отправка на исполнение”.  5а5. Возврат к шагу 1.  **5b. Сервис СД на запрос вернул код ответа 5xx.**  5b1. СОЗ формирует повторный запрос.  5b2. В случае неуспешного сценария, ошибка фиксируется в логе.  5b3. Сценарий завершен |

|  |
| --- |
| **ID:** UC-10 |
| **Название: Передача данных о доставке заказа клиенту** |
| **Участники: СОЗ, СД, Пользователь СД** |
| **Предусловия**:  1. В СОЗ создан и оплачен заказ. Статус заказа: “Принят к исполнению” 2. Заказ зарегистрирован в СД. Назначена дата доставки  **Триггер:** В ИС СДЭК статус заказа изменился на “Вручен клиенту” **Постусловия**: В СОЗ сохранена информация о доставке заказа клиенту, статус заказа изменился на “Доставлен”. |
| **Основной поток:**   1. СД в очереди сообщений публикует сообщение об изменении статуса заказа. 2. СОЗ отслеживает сообщение об изменении статуса заказа. 3. СОЗ выполняет авторизацию в сервисе СД. 4. Авторизация пройдена успешно. Сервис СД возвращает токен. 5. СОЗ отправляет в СД запрос на получение информации о заказе, идентификатор которого был получен в сообщении на шаге 2. 6. СД получает и обрабатывает запрос на предоставление информации о заказе. 7. СД передает в СОЗ информацию о заказе. 8. СОЗ обрабатывает полученную информацию, проверяет статус заказа. 9. СОЗ изменяет статус на “Доставлен”, сохраняет дату доставки заказа клиенту. 10. Сценарий завершен. |
| **Расширение:** **3а. Авторизация не пройдена.**3а1. Формируется повторный запрос авторизации.3а2. В случае неуспешного сценария, ошибка фиксируется в логе.3а3. Сценарий завершен **6a. Сервис СД на запрос вернул код ответа 5xx.**  6a1. СОЗ формирует повторный запрос.  6a2. В случае неуспешного сценария, ошибка фиксируется в логе.  6а3. Сценарий завершен. |

### Диаграмма последовательности

#### Оплата заказа

**

## Маппинг.Передача и преобразование данных Маппинг

## Описание данных, передаваемых из Системы продуктов в Службу обработки заказов

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Система-источник** | | | | **Система-приемник** | | | |
| **СП(С)** | | | | **СОЗ** | | | |
| ы **Объект/ атрибут** | |  | **Тип, значение** | **Объект/ атрибут** | |  | **Тип, значение** |
| Уникальный идентификатор продукта | ID | **\*** | \*UID, число\* | Идентификатор продукта | product\_id | \* | \*UID, число\* |
| Наименование продукта | Name |  | string(50) | Наименование продукта | product\_name |  | string(50) |
| Стоимость | Price | \* | string(50) | Стоимость продукта | product\_price | \* | string(50) |
| Количество на складе | Quantity\_stock |  | Число | Количество продукта для заказа | product\_quantity |  | Число |
| Параметры | Options |  | string(100) | Параметры продукта | product\_options |  | string(100) |
| Статус | State | \* | [активный, неактивный] | Статус продукта | product\_state | \* | [1, 0] |

|  |  |
| --- | --- |
| State | product\_state |
| активный | 1 |
| неактивный | 0 |

## Описание данных заказа, передаваемого СОЗ (С) в БС для расчета итоговой стоимости

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Система-источник** | | | | **Система-приемник** | | | |
| **СОЗ (С)** | | | | **БС** | | | |
| **Объект/ атрибут** | |  | **Тип, значение** | **Объект/ атрибут** | |  | **Тип, значение** |
| **Заказ** | **Order** | **=** | **\*Объект\*** | **Заказ** | **Order** | **=** | **\*Объект\*** |
| id заказа | id | = | \*UID, число\* | id заказа во внешней системе (СОЗ) | id \_ext\_Order | = | \* число\* |
| id заказа в СДЭК | cdek\_id | = | \*число\* | - | - | - | - |
| Дата заказа | Order\_data | = | \*Датавремя\* | - | - | - | - |
| Адрес доставки | delivery\_address | = | \*строка\* | Адрес доставки | address | = | <ссылка на Delivery\_adress> |
| Номер заказа | num | = | \*число\* | Номер заказа | number\_order | = | \*число\* |
| ID клиента | Recipient. recipient\_name | = | <ссылка на Recipient> | - | - | - | - |
| Стоимость | total\_sum | = | \*число\* | - | - | - | - |
| Статус | status\_id | = | [0, 1, 2, 3] | Статус обработки | State | = | [Принят, В обработке, Произведен расчет, Отправлен] |
| Тип доставки | delivery\_type | = | [курьером, до пункта выдачи] | - | - | - | - |
| - | - | - | - | id заказа в БС | id | = | \*UID, число\* |
| - | - | - | - | Дата регистрации заказа в системе БС | regdate | = | \*Датавремя\* |
| - | - | - | - | Стоимость заказа с учетом скидок | total\_sum | = | \*число\* |
| - | - | - | - | Стоимость доставки | delivery\_sum | = | \*число\* |
| **Позиции заказа** | **Order\_Items** | **=** | **\*Объект\*** | **Состав заказа** | **Order\_structure** | **=** | **\*Объект\*** |
| id позиции заказа | id\_Items | = | \*UID, число\* | id состава заказа | id\_structure | = | \*число\* |
| id заказа | <ссылка на Order> | = | \*число\* | id заказа | <ссылка на Order> | = | \*число\* |
| id продукта | id\_product | = | \*число\* | id товара | id\_goods | = | \*число\* |
| Количество | quantity | = | \*число\* | Количество | quantity | = | \*число\* |
| - | - | - | - | Цена | price | = | \*число\* |
| - | - | - | - | Валюта | currency | = | \*строка\* |
| - | - | - | - | Налог | tax | = | \*строка\* |
| - | - | - | - | Скидка | discount | = | \*число\* |

Преобразование данных

Преобразование значений атрибута status\_id

|  |  |
| --- | --- |
| СОЗ | БС |
| 0 | Принят |
| 1 | Отправлен |

## Описание технологии передачи данных

### REST API

### 7.1.1 Получение информации о заказе из СД

**Краткое описание метода**

Запрос предназначен для получения информации из СДЭК о доставке заказа клиенту.

**Метод запроса:** GET

**Строка запроса:** https://[host]:[port]/orders/?cdek\_number={ cdek\_number }

**Блок Authorization:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| access\_token | jwt-токен | string | + |

**Блок Headers:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| Content-Type | Тип передаваемых данных | application/json | + |

**Параметр URI**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| cdek\_number | ID заказа в СДЭК | integer | + |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Примеры ответа**  **200**  Content type  application/json  {   * "entity": {}, * "requests": [], * "related\_entities": []   } | Response Schema: application/json   |  |  | | --- | --- | | entity | object (OrderResponseDto)  Информация о заказе в рамках протокола 2.0 | | requests  required | Array of objects (RequestDto1)  Информация о запросах, относящихся к сущности. | | related\_entities | Array of objects (RelatedEntityDto)  Связанные сущности | |
| **Примеры ответа**  **400**  Content type  application/json  Копировать  Раскрыть всёЗакрыть всё  {   * "errors": [], * "warnings": []   } | Response Schema: application/json   |  |  | | --- | --- | | errors | Array of objects (ErrorDto)  Список ошибок | | warnings | Array of objects (WarningDto)  Список предупреждений | |

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Описание** | **Требуемые действия в системе** |
| 200 | Успешно | Проверить Выходные параметры (см. выше) |
| 400 | Некорректный запрос | Переслать администратору системы оповещение о произошедшей ошибке с указанием входных параметров запроса |

### **7.1.2 Запрос на регистрацию заказа в СД**

1. **Метод запроса:** POST
2. **Протокол:** https
3. **Строка запроса:** https://[host]:[port]/orders
4. **Блок Headers:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| **access\_token** | jwt-токен | string | + |
| Content-Type | Тип передаваемых данных | application/json | + |

1. **Входные параметры:**
2. **Формат запроса: application/json**
3. **Блок Body::**

|  |  |
| --- | --- |
| type | integer <int32>  Тип заказа.   * 1 - "интернет-магазин" (только для договора типа "Договор с ИМ"), * 2 - "доставка" (для любого договора). |
| additional\_order\_types | Array of integers <int32> unique [ items <int32 > ]  Дополнительные типы заказа.   * 4 - для Forward, * 6 - для "Фулфилмент. Приход", * 7 - для "Фулфилмент. Отгрузка", * 11 - для доставки в рамках одного офиса "Один офис (ИМ)" (при условии, что офис отправителя и получателя совпадают), * 14 - для CDEK.Shopping. |
| number | string  Номера заказа в ИС Клиента. Только для заказов "интернет-магазин". Может содержать только цифры, буквы латинского алфавита или спецсимволы (формат ASCII) |
| accompanying\_number | string  Номер сопроводительной накладной на товар (СНТ) |
| tariff\_code  required | integer <int32>  Код тарифа |
| comment | string  Комментарий к заказу |
| shipment\_point | string  Код ПВЗ СДЭК, на который будет производиться самостоятельный привоз клиентом. Обязательное поле, если заказ с тарифом "от склада". Не может использоваться одновременно с from\_location |
| delivery\_point | string  Код ПВЗ СДЭК, на который будет доставлена посылка. Обязательное поле, если заказ с тарифом "до склада" или "до постамата". Не может использоваться одновременно с to\_location |
| date\_invoice | string <date>  Дата инвойса. Обязательное поле, если заказ - международный с типом "интернет-магазин". Если поле заполнено, то заказ автоматически становится международным. |
| shipper\_name | string  Грузоотправитель. Обязательное поле, если заказ - международный с типом "интернет-магазин". Если поле заполнено, то заказ автоматически становится международным. |
| shipper\_address | string  Адрес грузоотправителя. Обязательное поле, если заказ - международный с типом "интернет-магазин". Если поле заполнено, то заказ автоматически становится международным. |
| delivery\_recipient\_cost | object (DeliveryRecipientCostRequestDto)  Доп. сбор за доставку, которую ИМ берет с получателя. Только для заказов "интернет-магазин". Для направлений Беларусь-Беларусь и РФ-Беларусь это поле игнорируется. |
| delivery\_recipient\_cost\_adv | Array of objects (DeliveryCostThresholdDto)  Доп. сбор за доставку (которую ИМ берет с получателя) в зависимости от суммы заказа (ДСД).  Только для заказов "интернет-магазин". Возможно указать несколько порогов. ДСД применима только с доп. услугой «ЧАСТИЧНАЯ ДОСТАВКА» - в случае, если получатель купил одну или несколько позиций, или получатель полностью отказался от выкупа товара.  Если при полном отказе от товара оплачивать доп. сбор не требуется, тогда при настройке порогов стоимости необходимо передавать дополнительную товарную позицию «Доставка» в количестве 1 шт. стоимостью 0 рублей, дополнительный сбор за доставку - 0 рублей и далее пороги ДСД.  Если при полном отказе от товара необходимо взять доп. сбор за доставку, то при настройке порогов в интеграции, необходимо указать дополнительную товарную позицию «Доставка» в количестве 1 шт. стоимостью 0 рублей, сумму доп.сбора за доставку с получателя необходимо задать при минимальном пороге стоимости, далее пороги ДСД, комментарии к заказу о необходимости вручения позиции "Доставка". |
| sender | object (SenderContactDto)  Отправитель. Не обязателен для заполнения, если заказ типа "интернет-магазин", обязателен, если заказ типа "доставка" |
| seller | object (SellerDto)  Реквизиты истинного продавца |
| recipient  required | object (RecipientContactDto)  Получатель |
| from\_location | object (RequestFromLocationDto)  Адрес отправления. Не может использоваться одновременно с shipment\_point. Обязателен, если заказ с тарифом "от двери". |
| to\_location | object (RequestToLocationDto)  Адрес получения. Не может использоваться одновременно с delivery\_point. Обязательное поле, если заказ с тарифом "до двери" |
| services | Array of objects (AdditionalServiceDto)  Дополнительные услуги |
| packages  required | Array of objects (PackageRequestDto)  Список информации по местам (упаковкам). Количество мест в заказе может быть от 1 до 255 |
| is\_client\_return | boolean  Признак клиентского возврата. Необходимо использовать, если прямой заказ доставлялся другой курьерской службой, либо по прямому заказу необходимо вернуть не все товары. Иначе необходимо использовать отдельный метод "Клиентские возвраты" (POST /v2/orders/{order\_uuid}/clientReturn) |
| developer\_key | string  Ключ разработчика |
| print | string  Enum: "WAYBILL" "BARCODE"  Тип печатной формы, которую необходимо сформировать по заказу |
| widget\_token | string  Токен CMS, содержащий дополнительные данные для заполнения данных о заказе |

**post**

**Примеры запроса**

Content type

application/json

Копировать

Раскрыть всёЗакрыть всё

{

* "type": 0,
* "additional\_order\_types": [],
* "number": "string",
* "accompanying\_number": "string",
* "tariff\_code": 0,
* "comment": "string",
* "shipment\_point": "string",
* "delivery\_point": "string",
* "date\_invoice": "2019-08-24",
* "shipper\_name": "string",
* "shipper\_address": "string",
* "delivery\_recipient\_cost": {},
* "delivery\_recipient\_cost\_adv": [],
* "sender": {},
* "seller": {},
* "recipient": {},
* "from\_location": {},
* "to\_location": {},
* "services": [],
* "packages": [],
* "is\_client\_return": true,
* "developer\_key": "string",
* "print": "WAYBILL",
* "widget\_token": "string"

}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Примеры ответа**   * **202**   Content type  application/json  {   * "entity": {}, * "requests": [], * "related\_entities": []   } | Response Schema: application/json   |  |  | | --- | --- | | entity | object (RootEntityDto)  Информация о сущности, над которой выполняется запрос | | requests  required | Array of objects (RequestDto1)  Информация о запросах, относящихся к сущности. | | related\_entities | Array of objects (RelatedEntityDto)  Связанные сущности | |
| **Примеры ответа**   * **400**   Content type  application/json  {   * "entity": { }, * "requests": [], * "related\_entities": []   } | Response Schema: application/json   |  |  | | --- | --- | | entity | object  Информация о сущности, над которой выполняется запрос | | requests  required | Array of objects (RequestDto1)  Информация о запросах, относящихся к сущности. | | related\_entities | Array of objects (RelatedEntityDto)  Связанные сущности | |

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Описание** | **Требуемые действия в системе** |
| 202 | Успешно | Проверить Выходные параметры (см. выше) |
| 400 | Некорректный запрос | Переслать администратору системы оповещение о произошедшей ошибке с указанием входных параметров запроса |

### **7.1.3 Запрос на получение базовой стоимости товара**

**Краткое описание метода**

Метод предназначен для получения стоимости товара в системе продуктов (склад) для расчета стоимости заказа с учетом скидок в биллинговой системе.

**Метод запроса:** **GET**

**Протокол:** https

**Строка запроса:** https://[host]:[port]/api/price

**Формат запроса:** application/json

**Блок Authorization:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| token | jwt-токен | string | + |

**Блок Headers:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| Content-Type | Тип передаваемых данных | application/json | + |

**Блок Body:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| ids | ID продукта (продуктов) в СП(С) | array | + |

**Пример запроса:**

curl --location --request GET 'https://[host]:[port]/api/price?ids=PR.MOB.CG22\_2,PR.MOB.3312&filter={"Field":"product\_id","product\_price"} ' \

--header 'Accept: application/json' \

--header 'Content-Type: application/json' \

--header 'Authorization: Bearer ed81a670b3cf6f235b8c0c51b709b4959c3c4232'

**Выходные параметры:**

**Формат ответа:** application/json

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** |
| product\_id | Идентификатор продукта | integer |
| product\_price | Стоимость продукта | String |

**Пример ответа:**

{

"products": [

{

"product\_id": "PR.MOB.CG22\_2",

"product\_price": "90000 р."

},

{

"product\_id": "PR.MOB.3312",

"product\_price": "12500 р."

}

]

}

**Пример ответа с ошибкой**

Добавить

**json-схема ответа**

{

"definitions": {},

"$schema": "http://json-schema.org/draft-07/schema#",

"$id": "https://example.com/object1671269582.json",

"title": "Root",

"type": "object",

"required": [

"products",

"product\_id",

"product\_price"

],

"properties": {

"products": {

"$id": "#root/products",

"title": "Products",

"type": "array",

"default": [],

"items":{

"$id": "#root/products/items",

"title": "Items",

"type": "object",

"required": [

"product\_id",

"product\_price"

],

"properties": {

"product\_id": {

"$id": "#root/products/items/product\_id",

"title": "Product\_id",

"type": "string",

"default": "",

"examples": [

"PR.MOB.CG22\_2"

],

"pattern": "^.\*$"

},

"product\_price": {

"$id": "#root/products/items/product\_price",

"title": "Product\_price",

"type": "string",

"default": "",

"examples": [

"90000 р."

],

"pattern": "^.\*$"

}

}

}

}

}

}

### **7.1.4 Запрос на передачу данных заказа в биллинговую систему**

**Краткое описание метода**

Метод предназначен для передачи данных заказа для расчета итоговой стоимости с учетом скидок в биллинговой системе.

**Метод запроса:** **POST**

**Протокол:** https

**Строка запроса:** https://[host]:[port]/calculateOrders

**Блок Headers:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| token | jwt-токен | string | + |
| Content-Type | Тип передаваемых данных | application/json | + |

**Входные параметры:**

**Формат запроса:** application/json

**Блок Body::**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Объект** | **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| Order | id | id заказа в БС | integer | - |
| id \_ext\_Order | id заказа во внешней системе (СОЗ) | integer | + |
| address | Адрес доставки | string | + |
| number\_order | Номер заказа | string | + |
| State | Статус обработки | string | + |
| regdate | Дата регистрации заказа в системе БС | datatime | - |
| total\_sum | cost | Стоимость заказа с учетом скидок | integer | - |
| currency | Валюта | string | - |
| delivery\_sum | cost | Стоимость доставки | integer | - |
| currency | Валюта | string | - |
| order\_items | id\_structure | id состава заказа | integer | + |
| id\_order | id связанного заказа | integer | + |
| id\_goods | id товара | integer | + |
| quantity | Количество | integer | + |
| price | Цена | integer | - |
| currency | Валюта | string | - |
| tax | Налог | string | - |
| discount | Скидка | integer | - |

**Пример запроса:**

curl --location --request POST 'https://[host]:[port]/calculateOrders\

--header '**Bearer Token**: 11a1111eb1b111ea1a1d111111111a11' \

- - header ‘Content-Type’\

-- body

{

"order": {

"id": null,

"id \_ext\_Order": "555",

"address": "Санкт-Петербург, Московский пр.,д.88",

"number\_order": "35236-323",

"State": "принят",

"regdate": null

},

"total\_sum": null,

"delivery\_sum": null,

"order\_items": [

{

"id\_structure": "1",

"id\_order": "555",

"id\_goods": "24552",

"quantity": 5,

"price": null,

"currency": null,

"tax": null,

"discount": null

},

{

"id\_structure": "2",

"id\_order": "555",

"id\_goods": "277777",

"quantity": 3,

"price": null,

"currency": null,

"tax": null,

"discount": null

}

]

}

**JSON схема запроса:**

{

"definitions": {},

"$schema": "http://json-schema.org/draft-07/schema#",

"$id": "https://example.com/object1671269785.json",

"title": "Root",

"type": "object",

"required": [

"order",

"total\_sum",

"delivery\_sum",

"order\_items"

],

"properties": {

"order": {

"$id": "#root/order",

"title": "Order",

"type": "object",

"required": [

"id",

"id\_ext\_Order",

"address",

"number\_order",

"State",

"regdate"

],

"properties": {

"id": {

"$id": "#root/order/id",

"title": "Id",

"type": "null",

"default": null

},

"id\_ext\_Order": {

"$id": "#root/order/id\_ext\_Order",

"title": "Id\_ext\_order",

"type": "string",

"default": "",

"examples": [

"555"

],

"pattern": "^.\*$"

},

"address": {

"$id": "#root/order/address",

"title": "Address",

"type": "string",

"default": "",

"examples": [

"Санкт-Петербург, Московский пр.,д.88"

],

"pattern": "^.\*$"

},

"number\_order": {

"$id": "#root/order/number\_order",

"title": "Number\_order",

"type": "string",

"default": "",

"examples": [

"35236-323"

],

"pattern": "^.\*$"

},

"State": {

"$id": "#root/order/State",

"title": "State",

"type": "string",

"default": "",

"examples": [

"принят"

],

"pattern": "^.\*$"

},

"regdate": {

"$id": "#root/order/regdate",

"title": "Regdate",

"type": "null",

"default": null

}

}

}

,

"total\_sum": {

"$id": "#root/total\_sum",

"title": "Total\_sum",

"type": "null",

"default": null

},

"delivery\_sum": {

"$id": "#root/delivery\_sum",

"title": "Delivery\_sum",

"type": "null",

"default": null

},

"order\_items": {

"$id": "#root/order\_items",

"title": "Order\_items",

"type": "array",

"default": [],

"items":{

"$id": "#root/order\_items/items",

"title": "Items",

"type": "object",

"required": [

"id\_structure",

"id\_order",

"id\_goods",

"quantity",

"price",

"currency",

"tax",

"discount"

],

"properties": {

"id\_structure": {

"$id": "#root/order\_items/items/id\_structure",

"title": "Id\_structure",

"type": "string",

"default": "",

"examples": [

"1"

],

"pattern": "^.\*$"

},

"id\_order": {

"$id": "#root/order\_items/items/id\_order",

"title": "Id\_order",

"type": "string",

"default": "",

"examples": [

"555"

],

"pattern": "^.\*$"

},

"id\_goods": {

"$id": "#root/order\_items/items/id\_goods",

"title": "Id\_goods",

"type": "string",

"default": "",

"examples": [

"24552"

],

"pattern": "^.\*$"

},

"quantity": {

"$id": "#root/order\_items/items/quantity",

"title": "Quantity",

"type": "integer",

"examples": [

5

],

"default": 0

},

"price": {

"$id": "#root/order\_items/items/price",

"title": "Price",

"type": "null",

"default": null

},

"currency": {

"$id": "#root/order\_items/items/currency",

"title": "Currency",

"type": "null",

"default": null

},

"tax": {

"$id": "#root/order\_items/items/tax",

"title": "Tax",

"type": "null",

"default": null

},

"discount": {

"$id": "#root/order\_items/items/discount",

"title": "Discount",

"type": "null",

"default": null

}

}

}

}

}

}

**Выходные параметры:**

**Формат ответа:** application/json

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| id | id заказа в БС | integer | + |
| error\_message | сообщение об ошибке | string | + |

**Код ответа (header):** 200 Ok

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Описание** | **Требуемые действия в системе** |
| 200 | Успешно | Проверить Выходные параметры (см. выше) |
| 400 | Некорректный запрос | Переслать администратору системы оповещение о произошедшей ошибке с указанием входных параметров запроса |
| 403 | Доступ запрещен | Отправить сообщение администратору системы оповещение о произошедшей ошибке с указанием входных параметров запроса |
| 500 | Внутренняя ошибка сервера | Отправить сообщение в тех. поддержку БС |

**Пример ответа:**

{

"id": "256457679696e332df"

}

Пример ответа с ошибкой

{

“error\_message”:”//”

**JSON схема ответа:**

{

"definitions": {},

"$schema": "http://json-schema.org/draft-07/schema#",

"$id": "https://example.com/object1671270035.json",

"title": "Root",

"type": "object",

"required": [

"id"

],

"properties": {

"id": {

"$id": "#root/id",

"title": "Id",

"type": "string",

"default": "",

"examples": [

"256457679696e332df"

],

"pattern": "^.\*$"

}

}

}

### 7.2 SOAP

#### 7.2.1 Краткое описание метода

Получить базовую стоимость товара для расчета стоимости заказа.

**Метод запроса:** POST

**Входные параметры:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| ids | ID продукта (продуктов) в СП(С) | ids[] | + |

**Пример запроса**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<filter>product\_id</filter>

<filter>product\_price</filter>

<product\_ids>i1</product\_ids>

<product\_ids>i2</product\_ids>

</root>

**XML-схема запроса**

**<?xml version="1.0" encoding="utf-8"?>**

**<!-- Created with Liquid Technologies Online Tools 1.0 (https://www.liquid-technologies.com) -->**

**<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">**

**<xs:element name="root">**

**<xs:complexType>**

**<xs:sequence>**

**<xs:element maxOccurs="unbounded" name="filter" type="xs:string" />**

**<xs:element maxOccurs="unbounded" name="product\_ids" type="xs:string" />**

**</xs:sequence>**

**</xs:complexType>**

**</xs:element>**

**</xs:schema>**

**Выходные параметры**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Описание** | **Тип данных** | **Обязательность** |
| product\_id | Идентификатор продукта | integer | + |
| product\_price | Стоимость продукта | String | + |

**Пример ответа**

<?xml version="1.0" encoding="UTF-8"?>

<root>

<products>

<product\_id>PR.MOB.CG22\_2</product\_id>

<product\_price>90000 р.</product\_price>

</products>

<products>

<product\_id>PR.MOB.3312</product\_id>

<product\_price>12500 р.</product\_price>

</products>

</root>

**XML-схема ответа**

**<?xml version="1.0" encoding="utf-8"?>**

**<!-- Created with Liquid Technologies Online Tools 1.0 (https://www.liquid-technologies.com) -->**

**<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">**

**<xs:element name="root">**

**<xs:complexType>**

**<xs:sequence>**

**<xs:element maxOccurs="unbounded" name="products">**

**<xs:complexType>**

**<xs:sequence>**

**<xs:element name="product\_id" type="xs:string" />**

**<xs:element name="product\_price" type="xs:string" />**

**</xs:sequence>**

**</xs:complexType>**

**</xs:element>**

**</xs:sequence>**

**</xs:complexType>**

**</xs:element>**

**</xs:schema>**

#### 7.2.2 Примеры запросов и ответов

**Пример SOAP-запрос**:

<soap:Envelope>

<soap:Body>

<m:CreatePayment xmlns:m="https://www.orders.org/">

<m:PaymentId>000152</m:PaymentId>

<m:ClientId>1037632967</m:ClientId>

<m:PaymentDate>2021-05-01</m:PaymentDate>

<m:PaymentSummRub>5000</m:PaymentSummRub>

</m:CreatePayment>

</soap:Body>

</soap:Envelope>

**SOAP-ответ:**

<soap:Envelope>

<soap:Body>

<xmlns:m="http://www.payment.org/" />

<m:CreatePaymentResponse>

<m: PaymentID>000152</m:PaymentID>

<m: PaymentStatus>Success</m:PaymentStatus>

</m:CreatePaymentPaymentResponse>

</soap:Body>

</soap:Envelope>

#### 7.2.3 SOAP схема запроса регистрации заказа в СД

<?xml version="1.0" encoding="utf-8"?>

<!-- Created with Liquid Technologies Online Tools 1.0 (https://www.liquid-technologies.com) -->

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="root">

<xs:complexType>

<xs:sequence>

<xs:element name="ext\_id" type="xs:unsignedShort" />

<xs:element name="order\_data">

<xs:complexType>

<xs:attribute name="null" type="xs:boolean" use="required" />

</xs:complexType>

</xs:element>

<xs:element name="delivery\_address" type="xs:string" />

<xs:element name="name" type="xs:string" />

<xs:element name="total\_sum">

<xs:complexType>

<xs:sequence>

<xs:element name="cost" type="xs:unsignedShort" />

<xs:element name="currency" type="xs:string" />

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="delivery\_sum">

<xs:complexType>

<xs:sequence>

<xs:element name="cost" type="xs:unsignedShort" />

<xs:element name="currency" type="xs:string" />

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="delivery\_type" type="xs:string" />

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>