

Smart POS

Документация по интеграции

1. Новая версия интеграции. Доступна с версии КР 2.25.1

Новая версия интеграции направленно на более гибкую обработку ошибок во время интеграционного процесса. Для обращения к новой версии интеграции нужно добавить ко всем запросам префикс /v2/. При этом в данный момент сохранена обратная совместимость со старой версией интеграции

Изменения в новой версии:

- Добавлена стандартная модель ответа от интеграции (см. п. 4.15). Предоставляет большую информативность при выполнении запросов
- Добавлены коды ошибок и соответствующие им текста (см. п. 4.15). Позволяет понять причину ошибки во время запроса
- Добавлены новые поля для запроса статуса транзакции, которые возвращают промежуточные статусы процесса оплаты или возврата и текст ошибки, в случае ее возникновения (см. п. 4.17). Позволяет более точно понять, на каком этапе транзакции находится Smart POS, и какая ошибка привела к неудачной транзакции.
- Новый метод получения актуального статуса. Позволяет актуализировать статус транзакции в случае, если во время ее выполнения произошла ошибка, не позволяющая понять результат транзакции (см. п. 4.4 и п. 4.18).

2. Термины

Термин	Описание
Kaspi Pay	Платежная система Kaspi
Smart POS	POS-терминал Kaspi Pay
Приложение Kaspi Pay	Версия прикладного ПО, установленное на Smart POS
Док-станция	Подставка для Smart POS, которая имеет порты подключения к питанию, порт LAN, порт для пинпада, порт USB для внешнего QR-дисплея
Интеграция со Smart POS	Реализация информационной связи между кассовым аппаратом и терминалом (Далее - Smart POS) с помощью TCP/IP соединения
Интеграция по LAN (Ethernet)	Реализация интеграции с Smart POS посредством ethernet сети
Интеграция по Wi-Fi	Реализация интеграции с Smart POS посредством Wi-Fi сети
Smart POS API	Http интерфейс для обмена данными, посредством которого строится интеграция со Smart POS
Токен	Ключ доступа к Smart POS API
ПО	Программное обеспечение

3. Основные требования и информация

- 3.1. Интеграция со Smart POS возможна только при подключении кассы и Smart POS к одной локальной сети.
- 3.2. Необходимо реализовать интеграцию со Smart POS только защищенной, приватной локальной сети партнера. Не следует реализовывать интеграцию с Smart POS через общедоступные локальные сети, этим могут воспользоваться злоумышленники.
- 3.3. Рекомендуем для интеграции использовать Ethernet (LAN), это необходимо для исключения задержки на прохождение данных между кассой и Smart POS.
- 3.4. В случае использования Wi-Fi для интеграции следует обеспечить надежность Wi-Fi сигнала. Бетонные стены, помехи от мощной электроаппаратуры, большое расстояние от точки доступа Wi-Fi, как и большое количество устройств в сети Wi-Fi может ухудшить Wi-Fi сигнал, следовательно, привести к проблемам интеграции.

- 3.5.** Интеграция со Smart POS возможна как в открытом виде через http, так и в защищенном виде через https с регистрацией кассы и аутентификацией по токену. Https дает дополнительную защиту от несанкционированного доступа к интеграции из локальной сети.
- 3.6.** Smart POS поддерживает 2 способа оплаты: с помощью Kaspi QR и с помощью карт платежных систем Visa, MasterCard. А также Smart POS поддерживает прием оплат с помощью платежных сервисов Google Pay, Apple Pay и Samsung Pay через NFC. При оплате клиенту доступно сразу оба способа оплаты, клиент самостоятельно выбирает удобный ему способ. (сканирует QR или прикладывает карту)
- 3.7.** В зависимости от выбранного клиентом способа оплаты будет меняться состав выходной (чековой) информации, получаемой кассой после успешной транзакции.
- 3.8.** Возврат оплаты в Smart POS должен быть произведен тем же способом, что и оплата. Нельзя вернуть оплату QR посредством карты, а карточную оплату посредством QR. Поэтому кассой должны быть зафиксированы параметры оплаты, для правильного проведения возврата.
- 3.9.** Требуется использовать актуальное ПО на Smart POS, в противном случае корректная работа интеграции не гарантируется.
- 3.10.** Обновление Smart POS проводится автоматически через сервисы Kaspi без дополнительного уведомления, если функциональность Smart POS не изменяется.
- 3.11.** Кассовая смена на Smart POS закрывается в автоматическом режиме.

4. Принцип работы

API интеграции на Smart POS имеет клиент-серверную архитектуру и работает по HTTP (HTTPS). В качестве клиента выступает кассовый аппарат, а сервером является Smart POS. API REST сервис Smart POS описан далее.

Доступ к Smart POS API интеграции осуществляется по IP адресу Smart POS либо по DNS-имени в локальной сети по порту 8080.

Кассовый аппарат отправляет запрос на подготовку транзакции (оплаты или возврата), получая в ответ уникальный идентификатор интеграционной операции processId.

В течении установленного времени касса периодически запрашивает статус текущей операции с помощью полученного processId. Рекомендуемый интервал запроса статуса – 1 секунда.

Smart POS производит транзакцию и обновляет статус процесса в памяти.

Касса получает результат операции в ответ на очередной запрос статуса. В случае успешно проведенной операции (статус «success») будут получены атрибуты транзакции, которые можно использовать для сохранения в отчет кассы и печати чека кассой.

В случае ошибки (статус «fail») будет получено краткое описание ошибки на английском, данная информация может понадобиться для отладки. Более подробное описание проблемы с транзакцией можно посмотреть непосредственно на экране Smart POS.

В момент времени возможно проводить только одну операцию, для начала новой операции нужно дождаться окончания предыдущей, о чем будет написано в описании ошибки, если попробовать совершить такую попытку.

Формат ответов от Smart POS – JSON.

В качестве параметра суммы передается целое число в тенге. «Тиын» не поддерживается в Smart POS.

Произведенные транзакции будут храниться в памяти Smart POS в течении суток, на случай обрыва связи с кассой. (Если касса не получила статус транзакции из-за обрыва связи между POS и кассой, то в течение суток можно запросить статус по «processId»)

4.1. Оплата

Для начала оплаты вызывается команда через URL «payment», в параметры передается сумма оплаты. Поскольку Smart POS поддерживает два способа оплаты: QR и карточную оплату, а клиент в процессе оплаты сам выбирает удобный способ, API запрос подготовки оплаты будет одинаковым для любого из этих методов.

При успешном запуске Smart POS перейдет в состояние ожидания оплаты, показав QR. В ответ на запрос придет «processId», который нужен для опроса статуса.

До окончания оплаты требуется периодически запрашивать статус оплаты.

Клиент выбирает способ оплаты сканируя QR или приложив карту.

В конце оплаты в ответ на запрос статуса вернется информация о транзакции, которая может быть использована для печати чека, для возврата оплаты и для кассового учета.

В зависимости от выбранного способа оплаты выходные чековые данные будут отличаться.

4.2. Возврат

В запросе на возврат нужно указать метод, которым была произведена оплата.

Возврат оплаты должен быть произведен тем способом, которым была произведена оплата. В противном случае клиент не сможет подтвердить возврат.

В связи с этим, после окончания оплаты требуется сохранить способ оплаты (поле «method») и идентификатор транзакции (поле «orderNumber» для QR оплаты и поле «RRN» для карточной оплаты соответственно). При вызове возврата этот идентификатор устанавливается в параметр «transactionId».

Возвраты возможно производить на полную сумму и на часть суммы.

Сумма возврата не может превышать доступную сумму возврата, которая зависит от сумм предыдущих частичных возвратов.

Как и при оплате Smart POS вернет «processId» операции, по которому до окончания возврата требуется периодически запрашивать статус.

Клиент должен подтвердить возврат отсканировав QR или приложив карту, в зависимости от способа оплаты. На экране будет отображен только требуемый способ подтверждения.

При возврате на карту при подтверждении это должна быть та же карта, которой производилась оплата. Если карта клиента, которой была произведена оплата перевыпущена, то возврат на карту через Smart POS будет невозможен, в этом случае нужно сделать возврат наличными.

В конце возврата в ответ на запрос статуса вернется информация о транзакции, которая может быть использована для печати чека и кассового учета.

В зависимости от способа возврата выходные чековые данные будут различные в зависимости от того, каким методом была произведена оплата.

4.3. Возврат без подтверждения клиента

Возврат без подтверждения клиента — это специальная функция, которая требуется для случаев, когда нет физической возможности клиента присутствовать при возврате. *(не рекомендуется для использования в виде основной функции возврата)*

В иных случаях должен использоваться возврат с подтверждением клиента.

Возврат без подтверждения возможен только через интеграцию. Напрямую из Smart POS сделать такой возврат невозможно.

Для возврата без подтверждения используется отдельный API запрос.

Кроме отсутствия подтверждения клиента, этот возврат функционально ничем не отличается от обычного возврата.

4.4. Актуализация статуса транзакции

Актуализация статуса транзакции требуется в случае, если во время выполнения транзакции произошла ошибка, которая не позволяет определить результат транзакции. Данный процесс позволяет определить результат транзакции и обновить его статус в интеграционном процессе. После успешного выполнения этот запрос также вернет все данные процесса, которые можно получить при запросе статуса процесса (статус, информация о транзакции в случае успешной транзакции, сообщение в случае ошибки при выполнении транзакции). Актуализировать можно только процессы со статусом “unknown”. В случае успешной актуализации этот процесс получит конечный статус (“fail” или “success”). Бывают моменты, когда запрос актуализации не вернет конечный статус, а статус процесса так и останется “unknown”. Это может произойти в случае, если при начале оплаты, после формирования QR произошла ошибка на Smart POS которая привела к перезапуску приложения или же кассир просто отменил оплату. Таким образом оплата так и будет со статусом “unknown” все время жизни QR (примерно 3 минуты). Затем уже транзакция будет считаться отмененной и можно будет актуализировать эту операцию на статус “fail”. Если же клиент успел отсканировать QR до того, как Smart POS перезапустился или кассир отменил оплату, то клиент может либо отменить оплату у себя на телефоне или же подтвердить ее. В таком случае можно будет сразу сделать запрос актуализации этого процесса и получить результат (“fail” - если клиент отменил оплату, и “success” - если подтвердил ее. Минимально допустимый интервал запроса актуализации – 10 секунд.

4.5. Печать чека

Печатать чек может как сам Smart POS, так и касса. Касса в результате успешной транзакции получит всю необходимую информацию для чека.

Опционально при старте транзакции указывается обработка чека на кассе, без печати чека на Smart POS через параметр API запроса owncheque. При значении owncheque=true печать на Smart POS при транзакции не будет выполняться.

Дизайн чека выполняется в соответствии с требованиями кассы.

Если символ «Т» в возвращаемом параметре «amount» не поддерживается кассой, то он может быть заменен на букву «Т» или надпись «тенге». Это стоит учесть при разработке интеграции на кассовом ПО (разработчиками кассового ПО).

4.6. Обязательные поля

Обязательные поля, которые должны быть на чеке.

Для QR оплаты:

- storeName – название магазина
- city – город
- address – адрес магазина
- status – текстовый статус транзакции
- amount – сумма
- date – дата
- bin – Бин организации
- terminalId – id терминала
- orderNumber – номер заказа qr

Для возврата по QR поля те же.

Для карточной оплаты:

- storeName – название магазина
- city – город
- address – адрес магазина
- status – текстовый результат транзакции
- amount – сумма
- date – дата
- cardMask – маска карты
- icc – эквайер карты
- bin – БИН организации
- terminalId – id терминала
- rrn – номер транзакции
- authorizationCode – код авторизации
- hostResponseCode – код ответа

Для карточного возврата:

- storeName – название магазина
- city – город
- address – адрес магазина
- status – текстовый результат транзакции
- amount – сумма
- date – дата
- cardMask – маска карты
- bin – Бин организации
- terminalId – id терминала
- rrn – номер транзакции
- authorizationCode – код авторизации

icc и hostResponseCode не передаются, т.к. в Trans Master используется manual refund.

4.7. Защита API интеграции

Поддерживается защита API при помощи SSL (HTTPS). В случае использования защищенного API, производится регистрация доступа кассы на Smart POS, и дальнейшая аутентификация запросов кассы по токenu. Функция защиты API включается индивидуально для каждого Smart POS.

Для включения требуется зайти в «Панель администратора», ввести пароль кассового администратора – ответственного за кассу со стороны партнера. В списке настроек нужно выбрать опцию «Защита интеграции».

Далее нужно нажать кнопку «Настроить доступ». Из кассы выполнить запрос «register», после чего появится запрос на доступ к API от кассы. Нажав кнопку «Разрешить», будет предоставлен доступ для кассы и в ответ будет отправлены параметры «token» и «revoke token». Токен нужно передавать в каждом запросе в HTTP Header для авторизации. Срок жизни токена 24 часа, после которых необходимо обновить токен через метод «revoke token».

Для обновления токена используется метод «revoke». Время истечения токена в поле «expirationDate», по которому можно определить время, когда нужно обновить токен. Можно обновить токен заранее, как только Smart POS получит запрос с новым токеном, старый будет считаться не валидным.

После истечения срока токена запрос будет завершен ошибкой HTTP 403 и соответствующим описанием.

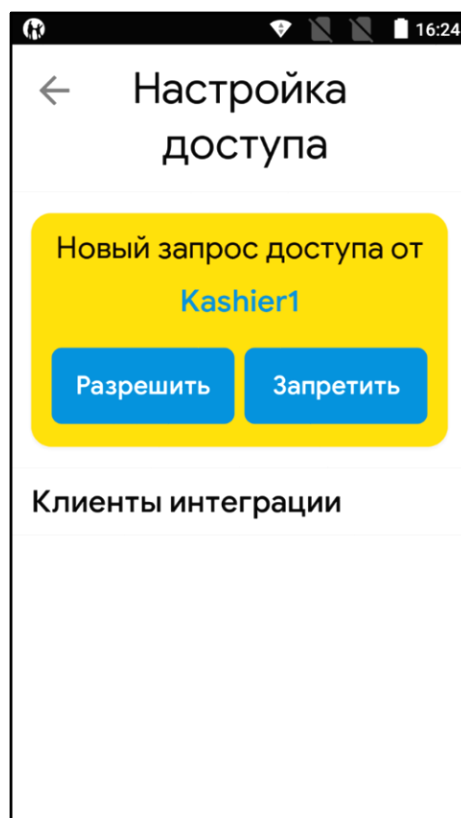


Рисунок 3.5.1

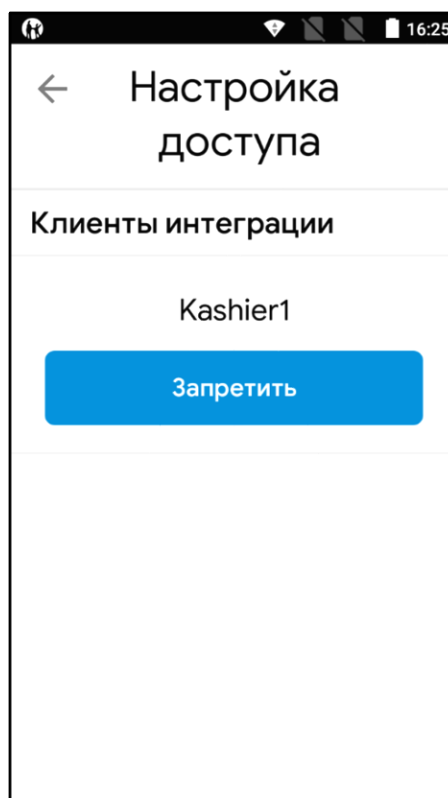


Рисунок 3.5.2

SSL сертификат выпущен доверенным CA для домена *.kaspiros.kz. Если в кассовом ПО настроена обязательная проверка сертификата, то требуется отправлять запросы по DNS имени Smart POS, который назначается в DNS-сервере локальной сети. Если DNS-сервер отсутствует, можно прописать DNS-имя в hosts-файл.

Если проверка сертификата не нужна, то можно обращаться к Smart POS через IP адрес, уровень защиты не изменится.

Порт при включенной защите так же 8080.

При включенной защите доступ по HTTP без SSL запрещен.

4.8. Блокировка интерфейса Smart POS при использовании

Ручной ввод через Smart POS можно заблокировать в панели администратора.

Эта функция позволит избежать проведение оплат/возвратов минуя кассу, соответственно исключить некорректный учет.

При этом вкладка «История» доступна, для просмотра истории транзакций в Smart POS.

Для включения функции ответственному за кассу со стороны Партнера, необходимо ввести пароль кассового администратора и в списке настроек включить опцию «Только интеграция».

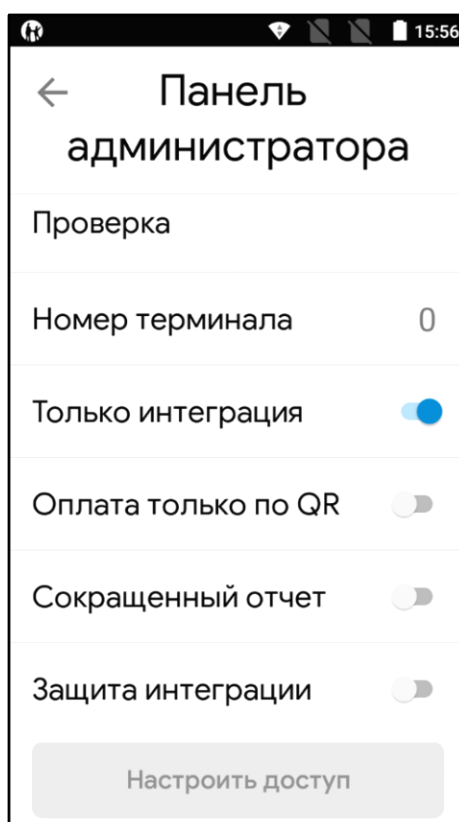


Рисунок 3.6.1

4.9. Возврат без использования интеграции

Возврат без использования интеграции, то есть напрямую через Smart POS.

Этот способ может потребоваться, если на Smart POS оплата успешно прошла, деньги у клиента списались, но касса не получила данные по транзакции. Либо в момент возврата интеграция не работала.

В таких случаях нужно отключить функцию «Только интеграция» через панель администратора и провести возврат напрямую через Smart POS, минуя кассу.

Метод возврата без использования интеграции: зайдите во вкладку «Возврат» в Smart POS и попросите клиента отсканировать QR или приложить карту, в зависимости от способа оплаты. В списке оплат клиента, на данной точке требуется выбрать нужную покупку и ввести сумму возврата на самом Smart POS.

После повторно необходимо провести оплату через интеграцию.

4.10. Удаленная оплата

Функция не поддерживается через интеграцию.

4.11. Подключение

IP-адрес для Smart POS должен быть статичным. Он может быть прописан партнером в Smart POS вручную, либо получен посредством DHCP-сервера привязкой статического IP-адреса к MAC-адресу Smart POS.

Порт интеграции на Smart POS 8080.

В приватной локальной сети в случае использования Firewall для ограничения доступа для корректной работы Smart POS в access list должны быть разрешен доступ к сервисам Kaspi Pay и сервисам регистрации ошибок Google. Если в приватной сети access list работает только с IP-адресами, то рекомендуется дать полный доступ в интернет для Smart POS.

Адрес	Порт	Описание
qrpay.kaspi.kz	443	сервис Kaspi Pay
pos.kaspi.kz	443	сервис Kaspi Pay
pos.kaspi.kz	9996	сервис обновления Kaspi Pay
spos-market.kaspi.kz	443	сервис обновления Kaspi Pay
spos-rkl.kaspi.kz	443	сервис Kaspi Pay
spos-info.kaspi.kz	443	сервис Kaspi Pay
Business.kaspi.kz	443	сервис Kaspi Pay
Poslog.kaspi.kz	443	сервис Kaspi Pay
*.crashlytics.com	80, 443	сервис регистрации ошибок
*.fabric.io	80, 443	сервис регистрации ошибок
crashlyticsreports-pa.googleapis.com	80, 443	сервис регистрации ошибок
firebasecrashlyticsymbols.googleapis.com	80, 443	сервис регистрации ошибок
*.1e100.net	80, 443	сервис регистрации ошибок
*.google.com	80, 443	сервис регистрации ошибок
*.jpush.cn	80, 443	сервис обновления системы
ota.m4bank.ru	80, 443	сервис обновления системы
supl.google.com	80, 443	сервис точного времени
north-america.pool.ntp.org	80, 443	сервис точного времени
time1.aliyun.com	80, 443	сервис точного времени
xtrapath1.izatcloud.net	80, 443	сервис геолокации
config.jpush.cn	443	сервис обновления системы

tsis.jpsh.cn	443	сервис обновления системы
user.jpsh.cn	443	сервис обновления системы
uploadlog.jpsh.cn	443	сервис обновления системы
api.jpsh.cn	443	сервис обновления системы
217.174.185.218	80	сервис обновления системы

Перед началом проверки интеграции стоит убедиться, что функционал Smart POS исправно работает без интеграции. Необходимо произвести оплату и возврат непосредственно через Smart POS.

Проверку интеграции можно делать с помощью браузера или тестовой программы Postman.



Рисунок 4.1

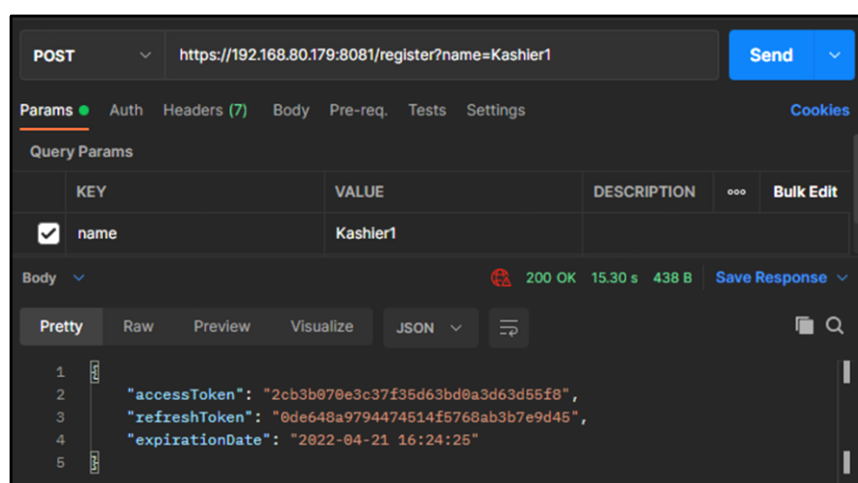


Рисунок 4.2

При возникновении проблем стоит убедиться, что доступ к сервисам Kaspi в приватной локальной сети настроен правильно.

4.12. Проверка связи

Можно проверить доступ к сервисам Kaspi, а также связь с кассой из экрана проверки связи в Smart POS. На экране оплаты в Smart POS нажмите на значок «Настройки»-«KaspiГид»-«О терминале», вкладка «Проверка» - «Проверить соединение». В случае проблем со связью можно выслать

фотографию данного экрана, а также экрана «Информация» специалистам Kaspi, для совместного решения.

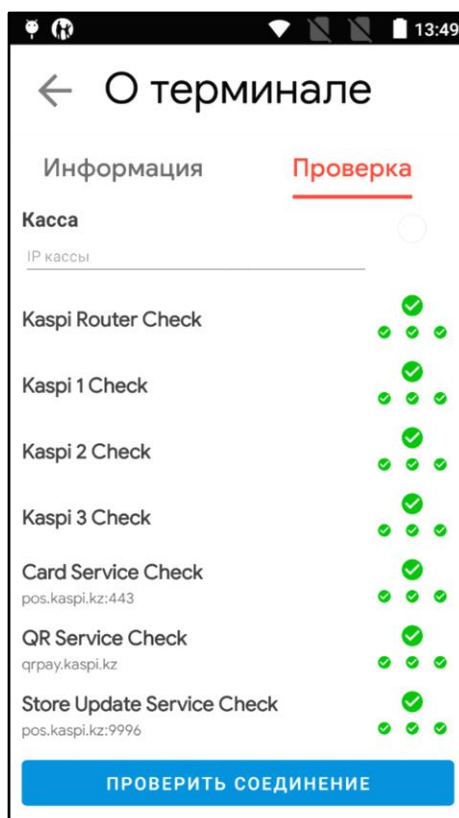


Рисунок 4.1.1

4.13. Подключение к Ethernet (LAN)

Требуется подключить кабель LAN к док-станции, подключить питание подставки и установить Smart POS на подставку.



Рисунок 4.2.1

Перейти в «Системные настройки» – «Ethernet», нажать на переключатель «Вкл». Проверить в статус баре отображение иконки док станции и LAN.

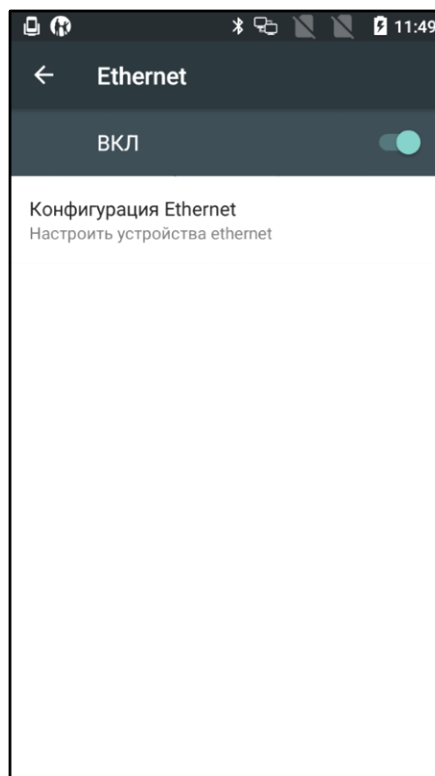


Рисунок 4.2.2

Нажать на «Конфигурацию Ethernet» и настроить параметры сети в соответствии с сетевыми требованиями приватной локальной сети.

В случае использования настроек DHCP-сервера можно закрепить IP-адрес Smart POS в соответствии с его MAC-адресом в сервере. Проверить закрепленный адрес можно в строке адреса.

SN	00022072642
IMEI	868133041734203
IP	10.28.24.211
Версия базовой полосы	BP01.002(SC20ESCR01A02)
SIM-карта	Карта не вставлена
WLAN MAC	90:BD:E6:A7:F2:B6
Адрес Bluetooth	90:BD:E6:A7:DB:34
Серийный номер	00022072642
MAC-адрес Ethernet	3c:39:e7:c6:3a:23
Шифрование устройства	Encrypted
Время загрузки	337:44:34

Рисунок 4.2.3

Если IP-адрес не будет закреплен в настройке DHCP-сервера, IP-адрес будет выдан в случайном порядке и связь с кассой будет недоступна, в этом случае нужно внести статический IP-адрес (следующий пункт).

Настройка оборудования Ethernet

Тип соединения

☒ DHCP

☐ Статический IP-адрес

IP-адрес

10.28.24.211

Длина префикса сети

Адрес DNS

Адрес шлюза

ОТМЕНИТЬ СОХРАНИТЬ

Рисунок 4.2.4

Нажать на «Конфигурация Ethernet». Выбрать «Статический IP-адрес» и заполнить адрес и параметры для статичного IP-адреса Smart POS.

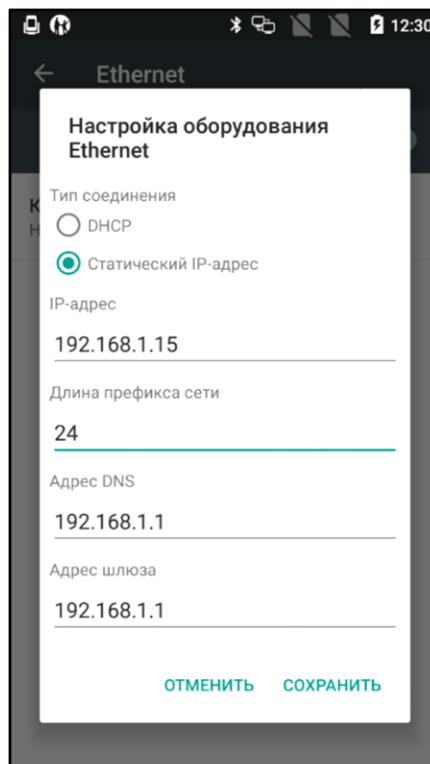


Рисунок 4.2.5

Конфигурацию самостоятельно вносит персонал, ответственный за локальную вычислительную сеть партнера. Сервис инженеры Kaspi могут оказать консультацию, помочь с активацией и проверкой работоспособности.

4.14. Подключение к Wi-Fi

Перейти в «Системные настройки» – «Wi-Fi», нажать на переключатель «Вкл». Выбрать из списка Wi-Fi сеть и ввести пароль для подключения. Проверить индикатор Wi-Fi.

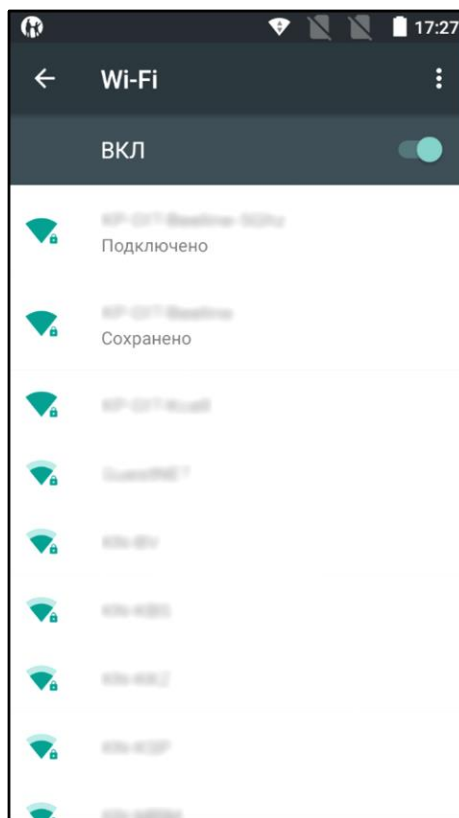


Рисунок 4.3.1

В случае использования настроек DHCP-сервера можно закрепить IP-адрес Smart POS в соответствии с его MAC-адресом. Проверить закрепленный адрес можно в строке адреса.

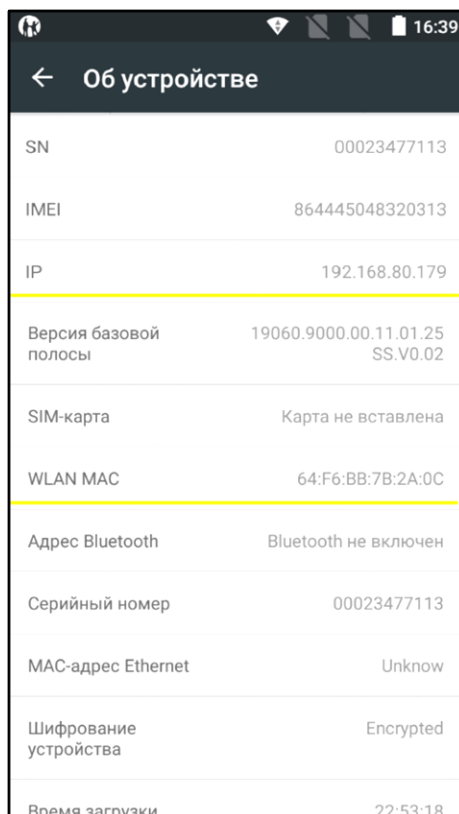


Рисунок 4.3.2

Длительным нажатием на выбранную сеть можно вызвать меню и «Изменить сеть».

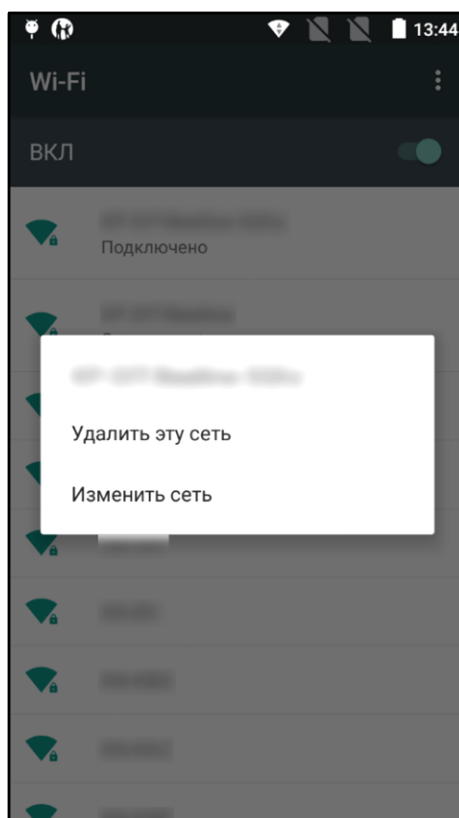


Рисунок 4.3.3

В открывшемся окне активировать чекбокс «Расширенные настройки» и пролистать вниз диалогового окна.

Если IP-адрес не будет закреплен за MAC-адресом в настройке DHCP-сервера, IP-адрес будет выдан в случайном порядке и связь с кассой будет недоступна, в этом случае нужно внести статический IP-адрес (следующий пункт).

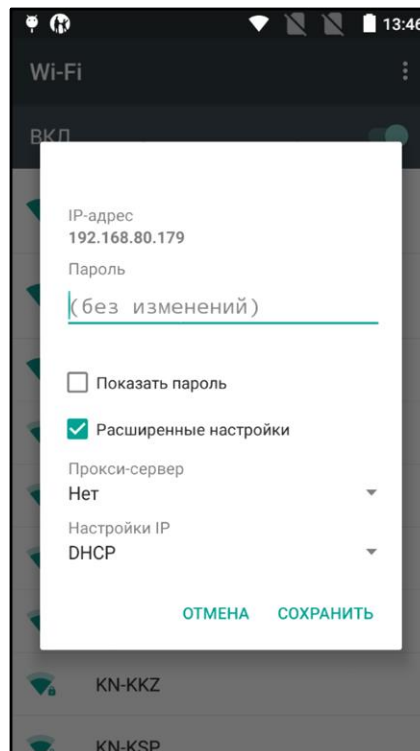


Рисунок 4.3.4

В меню «Настройки IP» выбрать «Статический IP-адрес» и заполнить адрес и параметры для статического IP-адреса Smart POS.

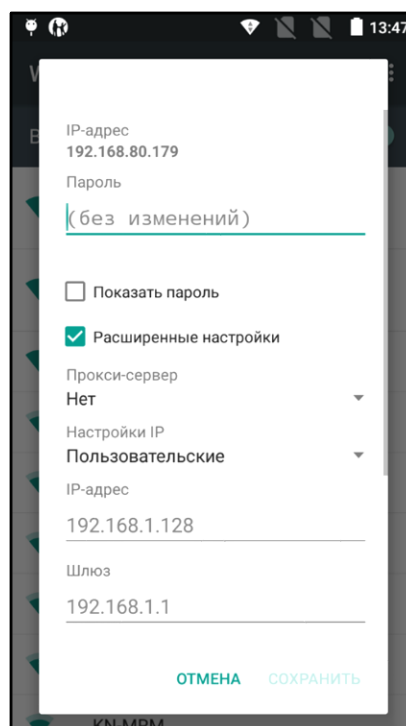


Рисунок 4.3.5

Конфигурацию для Wi-Fi самостоятельно вносит персонал, ответственный за локальную вычислительную сеть партнера. Сервис инженеры Kaspi могут оказать консультацию, помочь с активацией и проверкой работоспособности.

4.15. Описание API

Изменения в новой версии:

- Все запросы начинаются с префикса /v2/
- Изменена модель ответа при запросах. Все запросы возвращают единую модель, содержащую следующие поля: statusCode, data, errorText.

В примерах запроса будет использоваться адрес 192.168.80.4, адрес конкретного Smart POS будет зависеть от настройки в предыдущей главе.

В примерах запроса указан http запрос, для защищенного с помощью SSL варианта API соответственно нужно использовать https запрос, а в каждом HTTP запросе (кроме запроса «register») передавать Header «accesstoken» с токеном, полученном при запросе «register». Все выходные параметры являются обязательными, кроме указанных как «необязательные».

Ответ от API содержит следующие выходные параметры:

statusCode: Int – код ответа или код ошибки. Ниже приведены коды ошибок

data: Object – типизированное поле, содержащее выходные параметры конкретного запроса. В случае ошибочного запроса это поле не возвращается

errorText: String – текст ошибки. В случае успешного запроса это поле не возвращается

Коды ошибок, которые могут вернуться при запросе к API

httpCode	statusCode	errorText	Описание	Где используются
404	999	Invalid Url. Please, contact developers	Некорректная ссылка	Все методы
400	999	Missing param: {paramName}	Нет обязательного параметра в запросе. Отсутствующий параметр будет указан вместо {paramName}	
400	999	Param unvalidated: {paramName}. {cause}	Ошибка валидации параметра запроса. Ошибочный параметр будет указан вместо {paramName}. Описание ошибки валидации будет указано вместо {cause}	
400	106	Payments are blocked	Ошибка возникает при подключении Kaspi Kassa поверх интеграции или при блокировке транзакции.	
401	999	Authorization failed	Ошибка авторизации	
401	999	Session expires and need to be updated	Сессия истекла. Необходимо обновить сессию	

500	999	Internal Error. Please, contact developers	Внутренняя ошибка Smart POS, требуется снять лог с устройства специалистом Kaspi для определения причины ошибки.	
200	100	No internet	Ошибка интернета при выполнении процесса актуализации	/actualize
	101	Process not found	Не найдена транзакция с указанным processId. Возможно, произошла проблема при начале оплаты, либо срок хранения операции истек (срок хранения – 24 часа).	/status /actualize
	102	Unsupported transaction pay method	Метод, которым была оплачена транзакция (карта или QR) не поддерживается для актуализации	/actualize
	103	This transaction is unable to actualize	Статус этой транзакции уже нельзя актуализировать т.к. он уже финальный (“fail”, “success”)	/actualize
	104	Transaction not found	Транзакция, закрепленная за данным processId не найдена	/actualize
	105	Error acquired in service process	Ошибка во время выполнения запроса. Вместе с ошибкой возвращается поле data.message, содержащее текст ошибки	/actualize /register /revoke
	106	Payments are blocked	Оплата заблокирована	/payment
	107	Could not register transaction. Last operation was not completed.	Невозможно начать новый процесс оплаты/возврата, т.к. предыдущий процесс оплаты или возврата еще не завершен	/payment /refund /remains
	108	Could not register transaction.	Ошибка во время начала процесса оплаты/возврата	/payment /refund

		Something went wrong.		/remains
--	--	-----------------------	--	----------

4.16. Начало оплаты

Параметры запроса и ответа для начала оплаты с помощью QR или карты не будут отличаться. В случае успешного начала оплаты, в ответе вернется processId, который будет использован для запроса статуса текущего процесса оплаты.

Описание	Вызов начала оплаты			
HTTP Метод	GET			
URL	/payment			
Входные параметры	Наименование	Тип	Обязательный	Описание
	amount	Int	Да	Сумма покупки, целое числовое значение
	owncheque	Boolean	Нет	Обработка чека на кассе
Выходные параметры	processId: String – идентификатор процесса и используется для циклического запроса статуса текущей операции status: String – статус запрашиваемой операции.			
Пример	<p>http://192.168.80.4:8080/v2/payment?amount=10&owncheque=true</p> <pre>{ "data": { "processId": "1678090451700", "status": "wait" }, "statusCode": 0 }</pre>			
Пример ответа с ошибкой	<pre>{ "errorText": "Param unvalidated: amount. Must be greater 0", "statusCode": 999 }</pre>			

4.17. Запрос статуса транзакции

Изменения в новой версии:

- Добавлены новые выходные поля при запросе статуса транзакции:
 - message – содержит текст ошибки при ошибочном статусе (fail, unknown)
 - subStatus – промежуточный статус выполнения транзакции
- Добавлено новое значение для выходного параметра status при запросе статуса транзакции: unknown – означает, что результат транзакции неизвестен. Позволяет вызвать метод актуализации для этого процесса

Полученный в ответе на запрос начала транзакции processId используется как входной параметр запроса статуса.

Этот метод должен вызываться кассой циклически вплоть до окончания процесса. Окончанием процесса будет выходной параметр status, равный «success» либо «fail», либо «unknown». До окончания операции status будет «wait».

Также есть промежуточный статус subStatus, который показывает промежуточные этапы выполнения транзакции. Есть следующие значения: **Initialize**(инициализация процесса), **WaitUser**(ожидание действий клиента: скан QR или прикладывание карты), **WaitForQrConfirmation**(ожидание подтверждения оплаты через QR), **ProcessingCard**(выполнение карточной операции), **WaitForPinCode**(ожидание ввода клиентом пин-кода), **ProcessRefund**(выполнение возврата), **QrTransactionSuccess**(успешная оплата QR), **QrTransactionFailure**(ошибка оплаты QR), **CardTransactionSuccess**(успешная оплата картой), **CardTransactionFailure**(ошибка оплаты картой)

Выходные параметры для успешно завершенных запросов будут зависеть от выбранного клиентом способа оплаты, т.е. от платежной системы, с помощью которой клиент провел транзакцию. Так для QR транзакции уникальным идентификатором будет orderNumber, для карточной операции идентификатором будет RRN.

Для неуспешно завершенной операции в поле status будет «fail». В случае возникновения критической ошибки или отсутствия интернета во время оплаты или возврата, конечный статус транзакции будет «unknown», т.к. Smart POS не смог получить данные о результате транзакции.

Также среди выходных параметров будет поле message с текстом ошибки

Рекомендуемый интервал опроса статуса – одна секунда.

Тайм-ауты по умолчанию: на сканирование: 180 сек, на подтверждение оплаты: 60 сек.

**Важно: необходимо установить идентичные тайм-ауты на кассе со Smart POS.
(на сканирование QR и на подтверждение оплаты)**

Описание	Запрос статуса			
Метод	GET			
URL	/status			
Входные параметры	Наименование	Тип	Обязательный	Описание
	processId	string	Да	Идентификатор операции
Выходные параметры для QR	processId – идентификатор текущего процесса интеграции, status – статус операции, transactionId – строковый идентификатор успешной транзакции, addInfo: { IsOffer – транзакция по акции или нет (необязательный), ProductType – метод оплаты (необязательный), LoanTerm – срок кредита или акции (необязательный), LoanOfferName – название акции (необязательный, данное поле появляется в период действия акции/рассрочки) }, chequeInfo: { storeName – название магазина, city – город, address – адрес магазина, status – текстовый результат транзакции, amount – сумма, date – дата, bin – Бин организации, terminalId – id терминала, orderNumber – номер заказа qr, method – метод оплаты			

	<pre> }</pre>
Выходные параметры для карточной оплаты	<pre> processId – идентификатор текущего процесса интеграции, status – статус операции, transactionId – строковый идентификатор успешной транзакции, addInfo: { IsOffer – транзакция по акции или нет (необязательный), ProductType – метод оплаты (необязательный), LoanTerm – срок кредита или акции (необязательный), }, chequeInfo: { storeName – название магазина, city – город, address – адрес магазина, status – текстовый результат транзакции, amount – сумма, date – дата, cardMask – маска карты, icc – тип карты, bin – Бин организации, terminalId – id терминала, rrn – номер транзакции, authorizationCode – код авторизации, hostResponseCode – код ответа, pinEntered – пин код введен (необязательный), method – метод оплаты card } </pre>
Пример	<pre> http://192.168.80.4:8080/v2/status?processId=1643214345320 { "data": { "processId": "1678090708423", "status": "wait", "subStatus": "Initialize" }, "statusCode": 0 } </pre>

Пример ответа успешной оплаты	QR метод	Карточный метод
	<pre>{ "data": { "addInfo": { "LoanTerm": 0, "IsOffer": false, "ProductType": "Gold" }, "chequeInfo": { "address": "Конаева11, 6", "amount": "1 ₸", "bin": "960419351140", "city": "г. Алматы", "date": "06.03.23 14:20:11", "method": "qr", "orderNumber": "504711333", "status": "Покупка с Kaspi Gold. Одобрено", "storeName": "Kaspi Mark", "terminalId": "31452963", "type": "payment" }, "processId": "1678090708423", "status": "success", "subStatus": "QrTransactionSuccess", "transactionId": "504711333" }, "statusCode": 0 }</pre>	<pre>{ "data": { "addInfo": { "IsOffer": false, "LoanTerm": 0, "ProductType": "Gold" }, "chequeInfo": { "address": "Конаева11, 6", "amount": "1 ₸", "authorizationCode": "665425", "bin": "960419351140", "cardMask": "440043*****6389", "city": "г. Алматы", "date": "13.03.23 14:31:39", "hostResponseCode": "000", "icc": "Visa Debit", "method": "card", "rrn": "307208187011", "status": "Покупка с Kaspi Gold. Одобрено", "storeName": "Kaspi Mark", "terminalId": "31452963", "type": "payment" }, "processId": "1678696290407", "status": "success", "subStatus": "CardTransactionSuccess", "transactionId": "307208187011" }, "statusCode": 0 }</pre>
Пример ответа успешного возврата	<pre>{ "data": { "addInfo": { "IsOffer": false, "LoanTerm": 0, "ProductType": "Gold" }, "chequeInfo": { "address": "Конаева11, 6", "amount": "- 1 ₸", "bin": "960419351140", "city": "г. Алматы", "date": "06.03.23 14:21:40", "method": "qr", "orderNumber": "504711335", "status": "Возврат успешно совершен", </pre>	<pre>{ "data": { "addInfo": { "IsOffer": false, "LoanTerm": 0, "ProductType": "Gold" }, "chequeInfo": { "address": "Конаева11, 6", "amount": "- 1 ₸", "authorizationCode": "589719", "bin": "960419351140", "cardMask": "440043*****6389", "city": "г. Алматы", "date": "13.03.23 14:32:37", "hostResponseCode": "000", </pre>

	<pre> "storeName": "Kaspi Mark", "terminalId": "31452963", "type": "refund" }, "processId": "1678090915873", "status": "success", "subStatus": "QrTransactionSuccess", "transactionId": "504711335" }, "statusCode": 0 } </pre>	<pre> "icc": "", "method": "card", "rrn": "307208187011", "status": "Возврат успешно совершен", "storeName": "Kaspi Mark", "terminalId": "31452963", "type": "refund" }, "processId": "1678696351229", "status": "success", "subStatus": "CardTransactionSuccess", "transactionId": "307208187011" }, "statusCode": 0 } </pre>
Ответ с неудачно завершенной операцией	<pre> { "data": { "message": "Отмена покупки; Покупатель отменил покупку", "processId": "1678091002800", "status": "fail", "subStatus": "QrTransactionFailure" }, "statusCode": 0 } </pre>	
Ответ с неизвестным статусом транзакции	<pre> { "data": { "message": "Операция отменена", "processId": "1678092832132", "status": "unknown", "subStatus": "ProcessCancelled" }, "statusCode": 0 } </pre>	
Пример ответа с ошибкой	<pre> { "errorText": "Process not found", "statusCode": 101 } </pre>	

4.18. Актуализация статуса транзакции

Этот запрос позволяет актуализировать статус транзакции в случае возникновения ошибки во время ее выполнения. В таких случаях из запроса статуса транзакции (/v2/status) в поле status возвращается значение "unknown". В этом случае вы можете отправить запрос на актуализацию статуса. В остальных случаях ("fail", "success", "wait") этот запрос вернет ошибку. В случае успешного выполнения запроса в качестве выходных параметров он вернет те же, что и при обычном запросе статуса транзакции (/v2/status), но уже с обновленным статусом. В случае, если

этот запрос вернет также status “unknown”, то можно выполнить запрос повторно через некоторое время. В качестве входного параметра этот метод принимает processId, получаемый из запросов начала оплаты (/payment) или возврата (/refund или /remains). Максимальная частота запроса актуализации – раз в 10 секунд

Описание	Запрос статуса			
Метод	GET			
URL	/actualize			
Входные параметры	Наименование	Тип	Обязательный	Описание
	processId	Int	Да	Идентификатор операции
Выходные параметры	см. запрос статуса транзакции			
Пример	<p>http:// 192.168.80.4:8080/v2/actualize?processId=1678094695687</p> <pre>{ "data": { "message": "Отмена покупки; Покупатель не подтвердил покупку", "processId": "1678094695687", "status": "fail", "subStatus": "ProcessCancelled" }, "statusCode": 0 }</pre>			
Пример ответа со status: “unknown” из метода /v2/status	<pre>{ "data": { "message": "Операция отменена", "processId": "1678094695687", "status": "unknown", "subStatus": "ProcessCancelled" }, "statusCode": 0 }</pre>			

4.19. Возврат

Входной параметр «amount» – это сумма возврата. Она должна быть не более, чем доступная сумма возврата, которая в свою очередь зависит от суммы предыдущих возвратов.

Входной параметр method, который означает способ, которым клиент произвел оплату.

Возврат оплаты обязательно должен быть произведен тем способом, которым была произведена сама оплата.

Входной параметр «transactionId» – это уникальный идентификатор транзакции. Он должен быть взят из ответа статуса покупки в зависимости от метода оплаты, для QR оплаты – это поле «orderNumber», для карточной оплаты – поле «rrn».

Описание	Запрос начала возврата			
Метод	GET			
URL	/refund			
Входные параметры	Наименование	Тип	Обязательный	Описание
	amount	Int	Да	Сумма покупки

	method	String	Да	Метод операции. Возможные варианты – qr, card
	transactionId	String	Да	Идентификатор успешной транзакции
	owncheque	Boolean	Нет	Обработка чека на кассе
Выходные параметры	processId – идентификатор текущего процесса интеграции, status – статус операции, transactionId:string – строковый идентификатор успешной транзакции			
Пример	http://192.168.80.4:8080/v2/refund?method=card&amount=10&transactionId=202616353838 <pre>{ "data": { "processId": "1678092396883", "status": "wait" }, "statusCode": 0 }</pre>			
Ответ с ошибкой	<pre>{ "errorText": "Param unvalidated: method. Unknown payment method", "statusCode": 999 }</pre>			

4.20. Возврат без подтверждения клиента

Функционально возврат без подтверждения клиента идентичен обычному возврату, за исключением URL запроса.

Входной параметр «amount» – это сумма возврата. Она должна быть не более, чем доступная сумма возврата, которая в свою очередь зависит от суммы предыдущих возвратов.

Входной параметр method, который означает способ, которым клиент произвел оплату.

Возврат оплаты обязательно должен быть произведен тем способом, которым была произведена сама оплата.

Входной параметр «transactionId» – это уникальный идентификатор транзакции. Он должен быть взят из ответа статуса покупки в зависимости от метода оплаты, для QR оплаты – это поле «orderNumber», для карточной оплаты – поле «rrn».

Описание	Запрос начала возврата			
Метод	GET			
URL	/remains			
Входные параметры	Наименование	Тип	Обязательный	Описание
	amount	Int	Да	Сумма покупки
	method	String	Да	Метод операции. Возможные варианты – qr, card
	transactionId	Int	Да	Идентификатор успешной транзакции
	owncheque	Boolean	Нет	Обработка чека на кассе
Выходные параметры	processId – идентификатор текущего процесса интеграции, status – статус операции,			

	transactionId – строковый идентификатор успешной транзакции
Пример	<p>http://192.168.80.4:8080/v2/remains?method=card&amount=10&transactionId=202616353838</p> <pre>{ "data": { "processId": "1678092569387", "status": "wait" }, "statusCode": 0 }</pre>
Ответ с ошибкой	<pre>{ "errorText": "Param unvalidated: amount. Must be greater 0", "statusCode": 999 }</pre>

4.21. Регистрация кассы в защищенной API

Входной параметр «name» – имя кассы-клиента. Он также сохраняется и валидируется при обновлении токена.

Полученный параметр «token» передается в HTTP Header «accesstoken» в каждом запросе к кассе, в том числе в методе «revoke». Если токен не передан, то касса получит ошибку HTTP 401 Unauthorized.

По истечении срока жизни токена «expirationDate» при запросе с этим токеном касса получит ошибку HTTP 403 Forbidden.

В этом запросе Header «accesstoken» передавать не нужно, поскольку он и является источником токена.

Описание	Запрос регистрации кассы			
Метод	GET			
URL	/register			
Входные параметры	Наименование	Тип	Обязательный	Описание
	name	String	Да	Имя кассы
Выходные параметры	accessToken – токен авторизации, refreshToken – токен обновления, expirationDate – дата истечения срока токена			
Пример	<p>https:// 192.168.80.4:8080/v2/register?name=Kashier1</p> <pre>{ "data": { "accessToken": "99af370951a7b154a4ebd0c554d7eb24", "expirationDate": "Mar 7, 2023 15:02:56", "refreshToken": "d4a853dd320d179fde3271901c116bec" }, "statusCode": 0 }</pre>			

Пример ошибки регистрации	<pre>{ "data": { "message": "Register Error. Last register request not completed" }, "errorText": "Error acquired in service process", "statusCode": 105 }</pre>
Ответ с ошибкой	<pre>{ "errorText": "Missing param: name", "statusCode": 999 }</pre>

4.22. Обновление токена

В данном запросе так же, как и во всех других запросах к защищенному API нужно передавать Header «accesstoken» в котором будет просроченный токен.

Токен можно обновить раньше его даты истечения, старый токен будет очищен сразу, как только придет запрос с новым токеном.

Параметр «name» валидируется, должен быть тем же, как и при регистрации.

Обязательный параметр «refreshToken» полученный при регистрации кассы или последнем запросе «revoke».

Описание	Запрос обновления токена			
Метод	GET			
URL	/revoke			
Входные параметры	Наименование	Тип	Обязательный	Описание
	name	String	Да	Имя кассы
	refreshToken	String	Да	Токен обновления
Выходные параметры	accessToken – новый токен авторизации, refreshToken – новый токен обновления, expirationDate – новая дата истечения срока токена			
Пример	https://192.168.80.4:8080/v2/revoke?name=NewPOS&refreshToken=5a44d7db3ed03e50b262d569b3fcd728 <pre>{ "data": { "accessToken": "665495959cd859bce7b887cff45e79ba", "expirationDate": "Mar 7, 2023 15:15:50", "refreshToken": "343755b21c9388eda4a0a221d560410c" }, "statusCode": 0 }</pre>			

Пример ошибки обновления токена	<pre>{ "data": { "message": "Revoke Error. Wrong refresh token" }, "errorText": "Error acquired in service process", "statusCode": 105 }</pre>
Ответ с ошибкой	<pre>{ "errorText": "Param unvalidated: refreshToken. It's empty", "statusCode": 999 }</pre>

4.23. Информация о Smart POS

Запрос предназначен для проверки доступа в API Smart POS. Входных параметров нет. В ответе данные по устройству.

Описание	Запрос информации			
Метод	GET			
URL	/deviceinfo			
Входные параметры	Наименование	Тип	Обязательный	Описание
Выходные параметры	posNum – номер Smart POS, прописанный в панели администратора () serialNum – серийный номер, terminalId – Id терминала			
Пример	https://192.168.80.4:8080/v2/deviceinfo <pre>{ "data": { "posNum": "0", "serialNum": "00043010171", "terminalId": "31452963" }, "statusCode": 0 }</pre>			
Ответ с ошибкой	<pre>{ "errorText": "Invalid Url. Please, contact developers", "statusCode": 999 }</pre>			

Дополнительные функции:

- **Номер терминала**

Можно проставить нумерацию на каждом Smart POS, которая будет отображаться в верхней строке состояния на экране Smart POS.

Для указания номера необходимо на экране оплаты нажать на значок «Настройки» - «Панель администратора» - «Номер терминала» - укажите номер терминала – Ok.

- **Оплата только по QR**

Можно через «Панель администратора» включить или отключить карточный модуль. То есть если включить функцию «Оплата только по QR» будет невозможно оплатить картой.

- **Сокращенный отчет.**

Можно через «Панель администратора» включить функцию «Сокращенный отчет», которая позволит распечатывать или отправлять на email краткий отчет по продажам без детализаций.