



**Kaspi QR**

# Документация по интеграции v3.9

## 1. Введение

### 1.1. Термины

Термин	Описание
Kaspi Pay	Платежная система Kaspi QR
Организация	Организация, зарегистрированная в Kaspi Pay
Вендор	Поставщик программно-технического устройства/программного обеспечения для осуществления платежей в пользу организаций
Партнер	Вендор/Организация, интегрирующая Kaspi QR
QR-токен	Содержимое QR-кода (QR - картинки)
Сервер	Программно-аппаратная часть, отвечающая за функционирование внутренней части системы партнера.

### 1.2. Описание

**Важно!** При реализации партнер должен руководствоваться требованиями, предъявляемыми к оформлению:

Для киосков самообслуживания, вендинговых аппаратов, приложений курьеров и т.д.

- [Правила оформления элементов оплаты](#)
- [Элементы оплаты](#)

Для мобильных приложений и Web-сайтов:

- [Правила оформления элементов оплаты](#)
- [Элементы оплаты](#)

В инструкции представлены три схемы интеграции в зависимости от требований партнера:

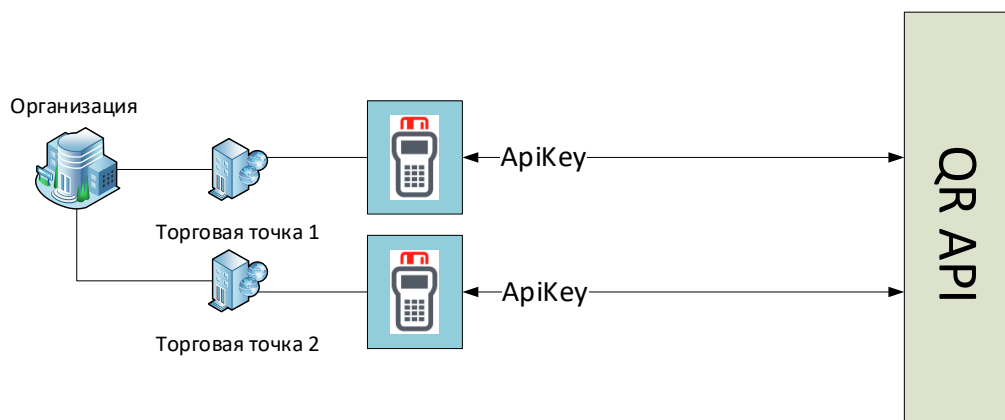
**Первая схема** Облегченный вариант доступа, в API нет метода возврата покупок.

Условия интеграции:

Авторизация запросов к API осуществляется по ApiKey. ⚠ ApiKey нельзя разглашать и передавать третьим лицам.

Возврат покупки осуществляется только через стандартный процесс возврата в мобильном терминале (приложение Kaspi Pay).

*Вариант реализации взаимодействия*



**Вторая схема** Стандартный вариант доступа, в API есть метод возврата покупки с участием покупателя.

Условия интеграции:

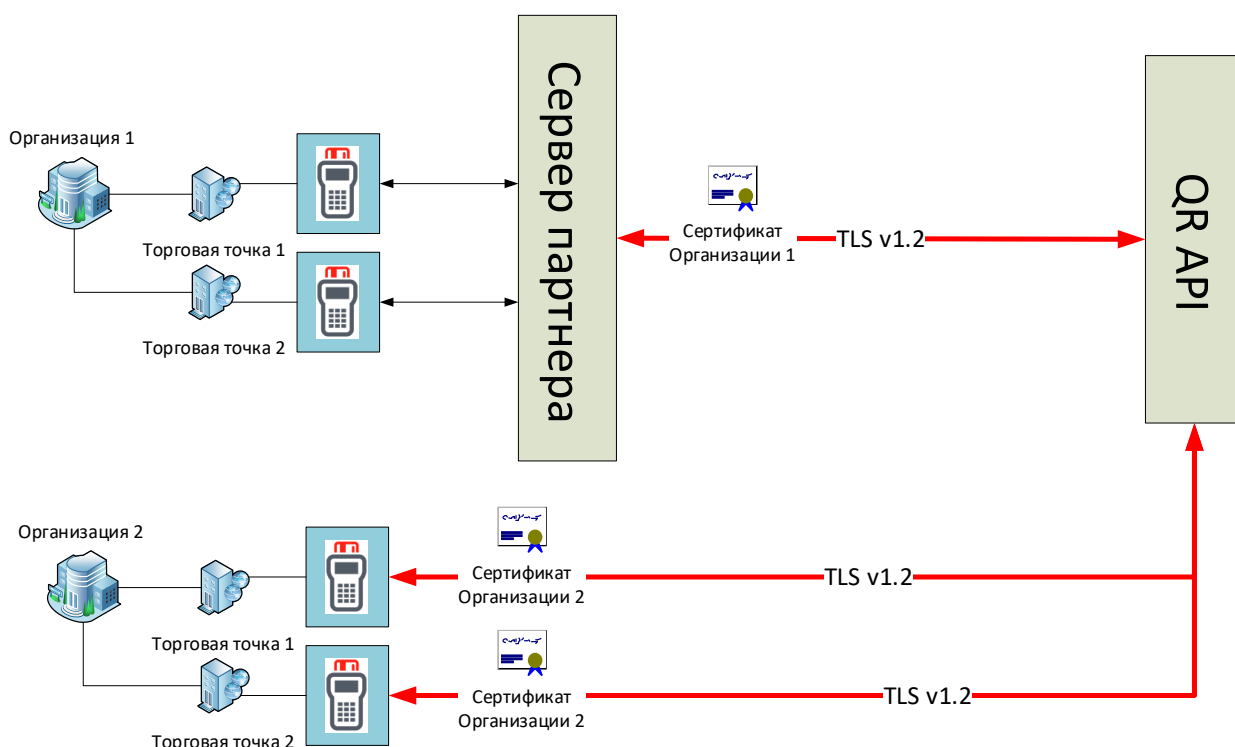
Авторизация запросов к API осуществляется по клиентскому сертификату.

Партнер может реализовать процесс возврат покупки в собственном приложении/устройстве. Обязательным условием процесса является сканирование покупателем QR-кода возврата.

Партнер может реализовать обращение к методам API как с использованием централизованного сервера, так и с устройств.

Безопасность соединения между партнером и платежной системой обеспечивается с помощью механизма двустороннего TLS (Mutual TLS, **TLS v1.2**).

#### *Варианты реализации взаимодействия*



**Третья схема** Вариант с усиленной безопасностью, в API есть метод возврата покупки без участия покупателя. Есть возможность выставления счета для удаленной оплаты.

Условия интеграции:

Партнер располагает сервером, который должен иметь статический IP-адрес.

Авторизация запросов к API осуществляется по клиентскому сертификату и IP-адресу.

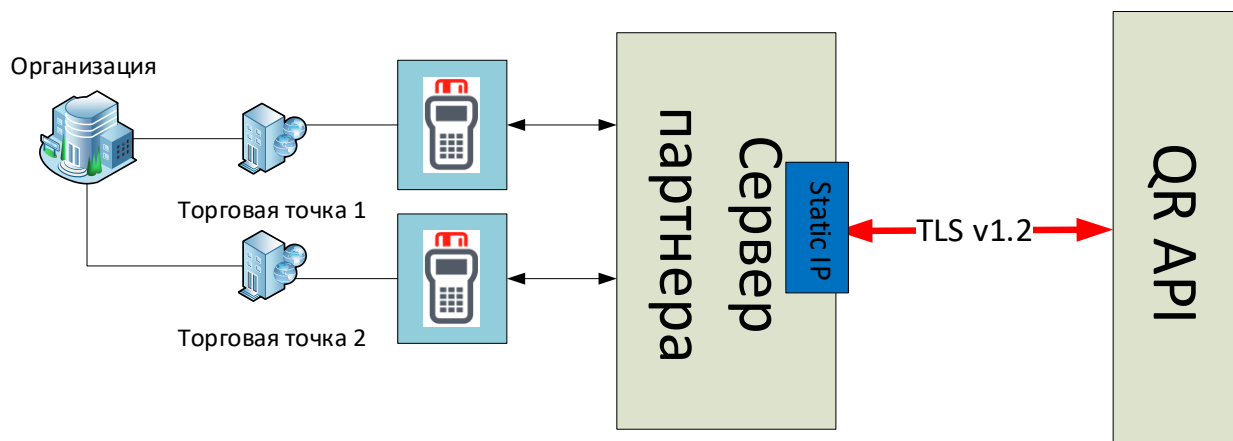
Партнер может реализовать процесс возврат покупки в собственном приложении/устройстве без участия покупателя.

Партнер может реализовать схему отправки запросов в API от лица другой организации со своим сертификатом.

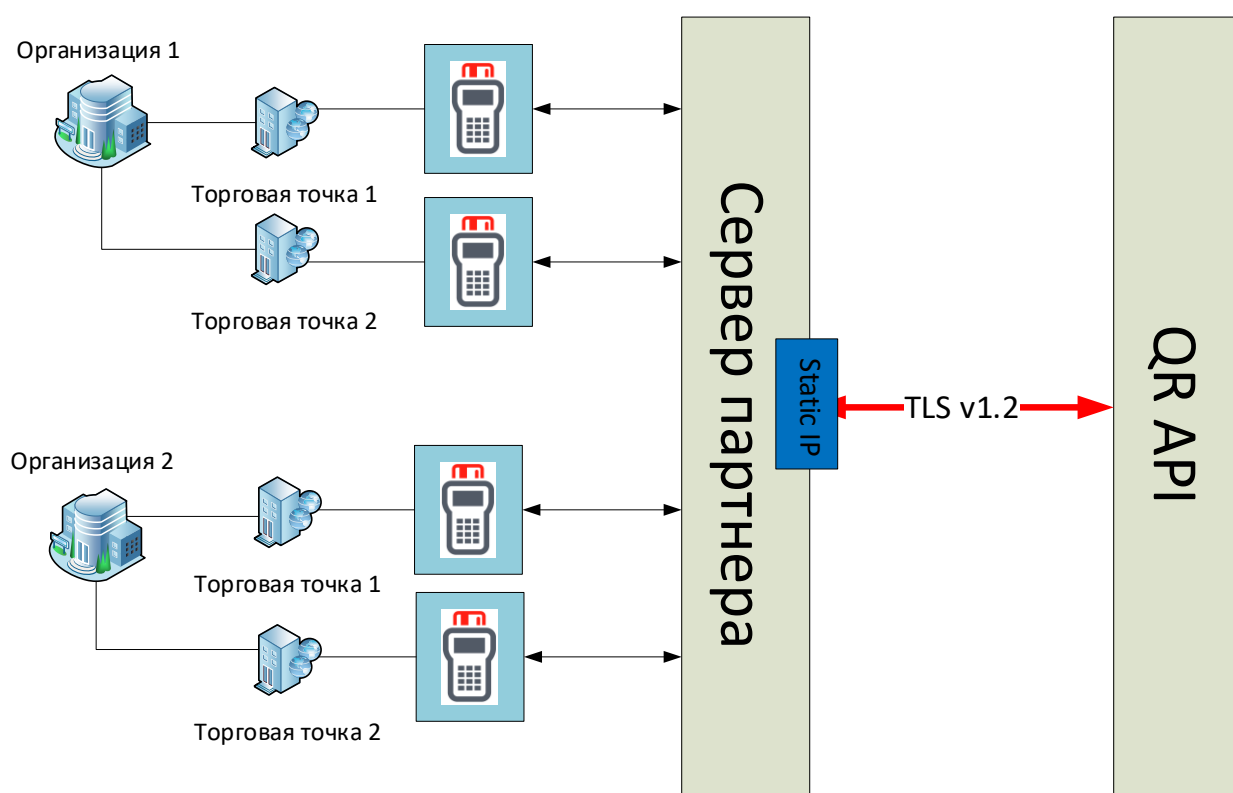
Для подключения партнеру необходимо передать **IP-адрес** своего сервера в банк. После этого партнеру выпускается и передается клиентский сертификат.

Безопасность соединения между партнером и платежной системой обеспечивается с помощью механизма двустороннего TLS (Mutual TLS, **TLS v1.2**) и фильтрацией запросов по IP-адресу партнера со стороны API.

*Вариант реализации процесса отправки запросов в API со своим сертификатом*



*Вариант реализации процесса отправки запросов в API от лица другой организации со своим сертификатом*



### 1.3. Описание вариантов авторизации

#### 1.3.1. ApiKey

Уникальный GUID закрепленный за организацией.

В каждый запрос к API нужно добавлять HTTP-заголовок "Api-Key" со значением ApiKey партнера.

#### ДЛЯ НАЧАЛА ТЕСТИРОВАНИЯ

Для тестирования партнеру передается тестовый ApiKey с префиксом test\_.

#### ДЛЯ ВЫХОДА В ПРОДАКШН

Для продакшена партнеру передается продовый **ApiKey** с префиксом prod\_.

### 1.3.2. Клиентский сертификат

Клиентский сертификат выпускается по запросу партнера. Срок действия сертификата **5 лет**. **Сертификат, приватный ключ и пароль приватного ключа** партнер использует для запросов к API. Безопасность соединения между партнером и платежной системой обеспечивается с помощью механизма двустороннего TLS (Mutual TLS, **TLS v1.2**). По истечению срока действия сертификата потребуется перевыпуск сертификата с последующей заменой старого на новый. Поэтому **при реализации клиентского приложения необходимо** предусмотреть возможность замены и дальнейшее использование обновленного сертификата без дополнительных доработок и привлечения разработчиков. Это же касается и **пароля приватного ключа**, который желательно запомнить в случае, если он прописывается в коде приложения. Лучше всего использовать конфигурационный файл, который используется в приложении, и при обновлении пароля обновить его в конфигурационном файле.

### ДЛЯ НАЧАЛА ТЕСТИРОВАНИЯ

Партнеру передается [ссылка](#) на архив, в котором собраны: тестовый сертификат, приватный ключ с указанием пароля, а также корневой и промежуточный сертификаты.

Подробнее о том, как выпустить продовый сертификат, установить и использовать, а также рекомендации к использованию и описаны в Приложении 1.

### 1.3.3. Сервер со статическим IP-адресом

Партнер располагает сервером, который должен иметь статический IP-адрес. Все запросы в API отправляются с этого сервера. Обращение к API осуществляется с использованием **клиентского сертификата**. Безопасность соединения между партнером и платежной системой обеспечивается с помощью механизма двустороннего TLS (Mutual TLS, **TLS v1.2**) и фильтрацией запросов по **IP-адресу** партнера со стороны API.

Для подключения партнеру необходимо передать **IP-адрес** своего сервера в банк. После этого партнеру выпускается и передается клиентский сертификат.

В случае наличия у партнера нескольких организации он может выпустить 1 сертификат и осуществлять платежи в пользу этих организаций.

## Первая схема – Облегченный вариант доступа

### 2. Первая схема – Облегченный вариант доступа

Я хочу интегрироваться по облегченной схеме, что мне для этого нужно?

- 1) Зарегистрироваться в приложении Kaspi Pay – [ссылка](#).
- 2) Настроить точки продаж и кассиров – [ссылка](#).
- 3) Отправить электронной почтой запрос в банк, для предоставления ApiKey.
- 4) Использовать полученный ApiKey при вызове методов API.

#### 2.1. Формат данных API

- Формат передачи данных: JSON (Content-Type: application/json)
- Формат даты: ISO 8601 (YYYY-MM-DDThh:mm:ss±h). Пример: 2020-08-09T18:31:42+06
- В каждый запрос к API нужно добавлять HTTP-заголовок “X-Request-ID” с уникальным значением (GUID/UID), генерируемый клиентом самостоятельно. Пример: X-Request-ID: da169655f65740d991b1bbfcd9802b28
- Все ответы имеют базовую структуру:

```
{
  "StatusCode": 0,
  "Message": "OK",
  "Data": {...}
}
```

Наименование	Тип	Обязательный	Описание
StatusCode	Int	Да	Код результата
Message	String	Нет	Сообщение результата, «OK» или «текст ошибки»
Data	Object	Нет	Объект данных, если имеется

Список StatusCode и их значения можно получить в Приложении 3.

#### 2.2. API List

##### 2.2.1. Регистрация кассы/устройства партнера

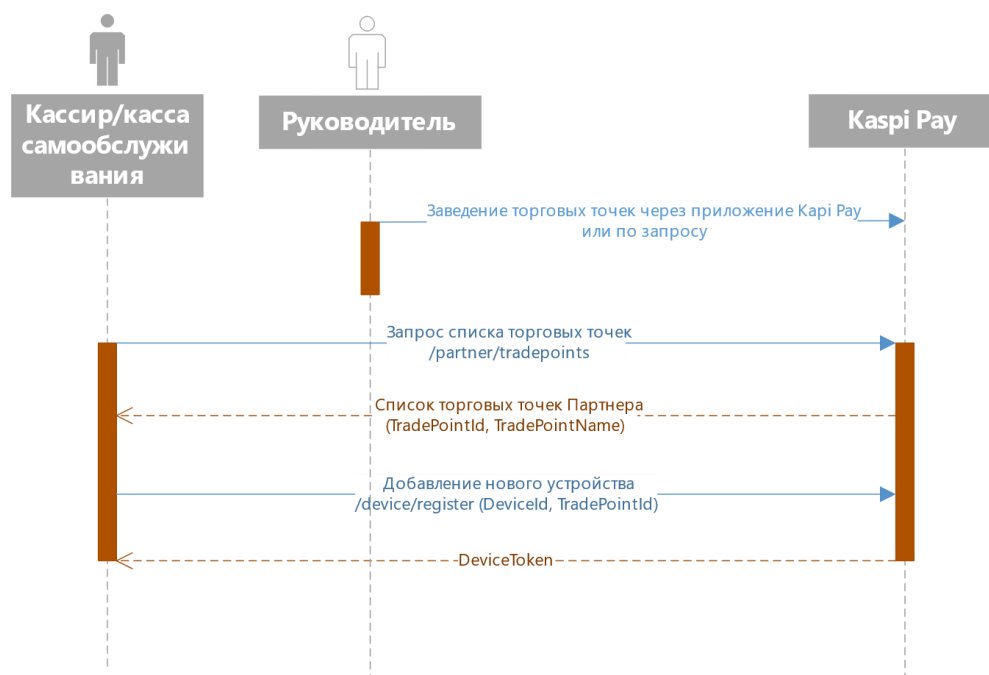


Рисунок 2.1 Регистрация устройства

## Первая схема – Облегченный вариант доступа

Перед регистрацией устройства нужно получить список торговых точек (см. п. 2.2.2) и выбрав нужную торговую точку вызвать метод добавления нового устройства (см. п. 2.2.3).

### 2.2.2. Список торговых точек

Описание	Получение списка торговых точек партнера для добавления устройства			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	GET			
Адрес	https://mtokentest.kaspi.kz:8543/r1/v01/partner/tradepoints			
Входные параметры				
Выходные параметры	TradePointId: Int64 – идентификатор торговой точки TradePointName: String – имя торговой точки			
Возможные StatusCode	0, -999, -10000, -14000002			
Пример	<p>https://mtokentest.kaspi.kz:8543/r1/v01/partner/tradepoints</p> <p>Ответ: {     "StatusCode": 0,     "Message": "OK",     "Data": [         { "TradePointId": 1, "TradePointName": "Магазин 1 (Ауэзова, 65)" },         { "TradePointId": 2, "TradePointName": "Магазин 2 (Жарокова, 48)" }     ] }</p>			

### 2.2.3. Регистрация устройства

Описание	Регистрация нового устройства в Kaspi Pay. В ответ на повторный вызов с теми же аргументами возвращается тот же результат.			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8543/r1/v01/device/register			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceId	String	Да	Идентификатор устройства в системе Партнера (⚠ Допустимы только латинские буквы, цифры, тире и нижнее подчеркивание)
	TradePointId	Int64	Да	Идентификатор торговой точки партнера в Kaspi Pay (см. п. 2.2.2).
Выходные параметры	DeviceToken: String – идентификатор зарегистрированного в Kaspi Pay устройства			
Возможные StatusCode	0, -999, -10000, -1503, -99000002, 990000026			

## Первая схема – Облегченный вариант доступа

Пример	<pre>https://mtokentest.kaspi.kz:8543/r1/v01/device/register  Запрос: {   "DeviceId": "GFC-456397",   "TradePointId": 12 }  Ответ: {   "StatusCode": 0,   "Message": "OK",   "Data": {     "DeviceToken": "2be4cc91-5895-48f8-8bc2-86c7bd419b3b"   } }</pre>
--------	--

### 2.2.4. Удаление/отключение устройства

Описание	Удаление/отключения устройства			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адреса	https://mtokentest.kaspi.kz:8543/r1/v01/device/delete			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceToken	Int64	Да	Идентификатор добавленного в Kaspi Pay устройства (см. п. 2.2.3)
Выходные параметры	Нет			
Возможные StatusCode	0, -999, -10000, -1501			
Пример	<pre>POST https://mtokentest.kaspi.kz:8543/r1/v01/device/delete  Запрос: {   "DeviceToken": "2be4cc91-5895-48f8-8bc2-86c7bd419b3b" }  Ответ: {   "StatusCode": 0 }</pre>			



## Первая схема – Облегченный вариант доступа

### 2.3. Оплата QR и по ссылке

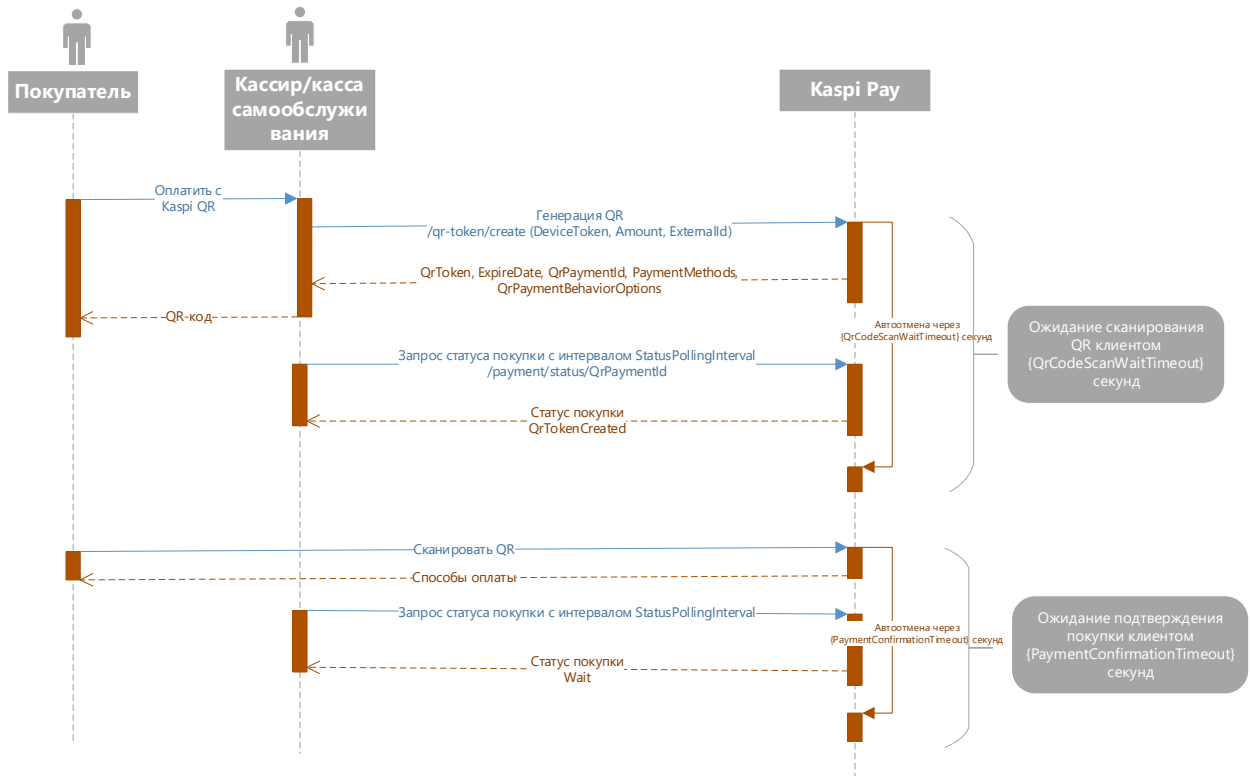


Рисунок 2.2 Процесс проведения покупки

Для того, чтобы начать процесс приема оплаты через QR-код, необходимо вызвать сервис создания QR-токена (см. п. 2.3.1). Вместе с QR-токеном в системе создается покупка. Полученный QR-токен нужно показать клиенту в виде изображения - QR-кода в приложении Партнера. Сразу после создания QR-токена, необходимо начать периодически запрашивать статус покупки (см. п. 2.3.3), с интервалом, указанным в выходном параметре сервиса создания QR-токена «StatusPollingInterval». Периодически запрашивать статус нужно до получения конечного статуса покупки, подробнее в Приложении 4. Также, QR-токен имеет ограниченное время жизни, которое состоит максимум из двух этапов:

**1) Ожидание сканирования клиентом.** Сразу после создания QR-токена, покупка имеет статус **QrTokenCreated**. Приложение Партнера должно зафиксировать «текущее время» + `QrCodeScanWaitTimeout` и если по истечении этого времени статус покупки не изменится и останется **QrTokenCreated**, то необходимо убрать отображение QR-кода и показать экран ошибки, т.к. QR-токен уже не действителен и сканирование QR-кода приведет к ошибке на стороне клиента.

Если зафиксированное время не истечет и статус покупки изменится на:

- **Wait**, то это означает, что клиент успешно отсканировал QR-код и желательно сменить экран отображения QR-кода на экран «ожидание подтверждения покупки» и перейти к следующему этапу.

- **Error**, то необходимо показать экран ошибки.

**2) Ожидание выбора счета оплаты и подтверждения покупки клиентом.** После успешного сканирования QR-кода клиентом, покупка имеет статус **Wait**. Приложение Партнера должно зафиксировать «текущее время» + `PaymentConfirmationTimeout` и если по истечении этого времени статус покупки не изменится и останется **Wait**, то необходимо сменить экран «ожидание подтверждения покупки» на экран ошибки. Покупка автоматически отменится и деньги вернутся клиенту. Если зафиксированное время не истечет и статус покупки изменится на:

## Первая схема – Облегченный вариант доступа

- **Processed**, то показать экран успешной покупки.
- **Error**, то необходимо показать экран ошибки.

### 2.3.1. Запрос на создание QR

При вызове метода создания QR-токена в ответе приходят следующие параметры, которые необходимо использовать при реализации приложения:

1. **PaymentMethods** – список способов оплаты, доступных Партнеру. [“Gold”, “Red”, “Loan”]  
Использовать при отображении с QR доступных способов оплаты.
2. **QrPaymentBehaviorOptions** – настройки временных интервалов для процессов QR-оплаты, использовать в логике работы приложения: получения статусов, смены экранов, вывода ошибок:
  - 2.1. **StatusPollingInterval** – интервал в секундах, через который необходимо периодически вызывать метод получения статуса операции
  - 2.2. **QrCodeScanWaitTimeout** – таймаут (в секундах) сканирования QR-кода (до получения статуса Wait).
  - 2.3. **PaymentConfirmationTimeout** – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)

Описание	Создание QR-токена			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8543/r1/v01/qr/create			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceToken	String	Да	Идентификатор добавленного в Kaspi Pay устройства
	Amount	Double	Да	Сумма покупки
	ExternalId	String	Нет	Идентификатор покупки в системе Партнера
Выходные параметры	QrToken: String – QR-токен ExpireDate: DateTime – дата и время истечения срока действия QR-токена. Сканирование QR-токена с истекшим сроком действия приводит к ошибке. QrPaymentId: Int64 – идентификатор покупки: используется для получения статуса покупки, возврата покупки. PaymentMethods: String[] – список способов оплаты, доступных Партнеру QrPaymentBehaviorOptions – настройки временных интервалов для процессов QR-оплаты: StatusPollingInterval – интервал в секундах, через который необходимо периодически вызывать статус операции (см.п. 2.3.3) QrCodeScanWaitTimeout – таймаут (в секундах) сканирования QR-кода (до получения статуса Wait). PaymentConfirmationTimeout – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)			
Возможные StatusCode	0, -999, -10000, -1501, -1502, -990000002, 9900000018, 9900000026, 9900000028, 9900000033, 9900000033			
Пример	https://mtokentest.kaspi.kz:8543/r1/v01/qr/create Запрос: { “DeviceToken”: “2be4cc91-5895-48f8-8bc2-86c7bd419b3b”, “Amount”: 200.00, “ExternalId”: “15” }			

## Первая схема – Облегченный вариант доступа

	<pre>}  Ответ: {   "StatusCode": 0,   "Message": "OK",   "Data": {     "QrToken": "51236903777280167836178166503744993984459",     "ExpireDate": "2020-01-16T10:30:00+06",     "QrPaymentId": 15,     "PaymentMethods": [ "Gold", "Red", "Loan" ],     "QrPaymentBehaviorOptions": {       "StatusPollingInterval": 5,       "QrCodeScanWaitTimeout": 180,       "PaymentConfirmationTimeout": 65     }   } }</pre>
--	---

### 2.3.2. Запрос на создание ссылки на оплату

Данный сервис используется для приема оплаты через ссылку. В мобильном приложении партнера необходимо вызвать сервис создания ссылки на оплату (2.3.2). Полученную ссылку нужно использовать в мобильном приложении Партнера. После того, как клиент активирует ссылку (перейдет по ссылке) в мобильном приложении партнера будет произведено переправление в **мобильное приложение Kaspi.kz** (должно быть установлено) на стороне клиента. Если мобильное приложение Kaspi.kz не установлено у клиента, то активация ссылки приведет к открытию главной страницы сайта Kaspi.kz в мобильном браузере.

Сразу после создания ссылки, необходимо начать периодически запрашивать статус покупки (см. п. 2.3.3), с интервалом, указанным в выходном параметре сервиса создания ссылки «StatusPollingInterval». Периодически запрашивать статус нужно до получения конечного статуса покупки, подробнее в Приложении 4. Также, ссылка имеет ограниченное время жизни, которое состоит максимум из двух этапов:

- 1) **Ожидание активации ссылки клиентом.** Сразу после создания ссылки, покупка имеет статус **QrTokenCreated**. Приложение Партнера должно зафиксировать «текущее время» + QrCodeScanWaitTimeout и если по истечении этого времени статус покупки не изменится и останется **QrTokenCreated**, то необходимо убрать элементы связанные с ссылкой, т.к. ссылка уже не действительна и переход по ней приведет к ошибке на стороне клиента.

Если зафиксированное время не истечет и статус покупки изменится на:

- **Wait**, то это означает, что клиент успешно активировал ссылку (перешел по ссылке).
- **Error**, то необходимо показать экран ошибки.

- 2) **Ожидание выбора счета оплаты и подтверждения покупки клиентом.** После успешной активации ссылки клиентом, покупка имеет статус **Wait**. Приложение Партнера должно зафиксировать «текущее время» + PaymentConfirmationTimeout и если по истечении этого времени статус покупки не изменится и останется **Wait**, то необходимо показать экран ошибки. Покупка автоматически отменится и деньги вернутся клиенту. Если зафиксированное время не истечет и статус покупки изменится на:

- **Processed**, то показать экран успешной покупки.
- **Error**, то необходимо показать экран ошибки.

При вызове метода создания ссылки на оплату в ответе приходят следующие параметры, которые необходимо использовать при реализации приложения:

## Первая схема – Облегченный вариант доступа

1. **PaymentMethods** – список способов оплаты, доступных Партнеру. [“Gold”, “Red”, “Loan”]
2. **PaymentBehaviorOptions** – настройки временных интервалов для процессов оплаты, использовать в логике работы приложения: получения статусов, смены экранов, вывода ошибок:
  - 2.1. **StatusPollingInterval** – интервал в секундах, через который необходимо периодически вызывать метод получения статуса операции
  - 2.2. **LinkActivationWaitTimeout** – таймаут (в секундах) активации ссылки (до получения статуса Wait).
  - 2.3. **PaymentConfirmationTimeout** – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)

Описание	Создание ссылки на оплату			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8543/r1/v01/qr/create-link			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceToken	String	Да	Идентификатор добавленного в Kaspi Pay устройства
	Amount	Double	Да	Сумма покупки
	ExternalId	String	Нет	Идентификатор покупки в системе Партнера
Выходные параметры	PaymentLink: String – ссылка на оплату ExpireDate: DateTime – дата и время истечения срока действия ссылки. Переход по ссылке с истекшим сроком действия приводит к ошибке. PaymentId: Int64 – идентификатор покупки: используется для получения статуса покупки, возврата покупки. PaymentMethods: String[] – список способов оплаты, доступных Партнеру PaymentBehaviorOptions – настройки временных интервалов для процессов оплаты: StatusPollingInterval – интервал в секундах, через который необходимо периодически вызывать статус операции (см.п. 2.3.3) LinkActivationWaitTimeout – таймаут (в секундах) активации ссылки (до получения статуса Wait). PaymentConfirmationTimeout – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)			
Возможные StatusCode	0, -999, -10000, -1501, -1502, -99000002, 990000018, 990000026, 990000028, 990000033			
Пример	https://mtokentest.kaspi.kz:8543/r1/v01/qr/create-link Запрос: <pre>{   "DeviceToken": "2be4cc91-5895-48f8-8bc2-86c7bd419b3b",   "Amount": 200.00,   "ExternalId": "15" }</pre> Ответ: <pre>{   "StatusCode": 0,   "Message": "OK", }</pre>			

## Первая схема – Облегченный вариант доступа

	<pre>"Data": {   "PaymentLink":   "https://pay.kaspi.kz/pay/59217969256592032202439955771435979527 677",   "ExpireDate": "2020-01-16T10:30:00+06",   "PaymentId": 15,   "PaymentMethods": [ "Gold", "Red", "Loan" ],   "PaymentBehaviorOptions": {     "StatusPollingInterval": 5,     "LinkActivationWaitTimeout": 180,     "PaymentConfirmationTimeout": 65   } }</pre>
--	---

### 2.3.3. Получение статуса операции

Описание	Получение текущего статуса покупки. Список возможных статусов смотрите в Приложении 4. Статус операции необходимо запрашивать с периодичностью каждые n-секунд, указанной в значении StatusPollingInterval, полученном при генерации QR. Статус покупки необходимо запрашивать до тех пор, пока не получен конечный статус, или не истекло время жизни токена. Нормальный жизненный цикл покупки: QrTokenCreated -> Wait -> Processed -> Error			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	GET			
Адрес	https://mtokentest.kaspi.kz:8543/r1/v01/payment/status/{QrPaymentId}			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	QrPaymentId	Int64	Да	Идентификатор покупки
Выходные параметры	Status: String – статус операции (см. Приложение 4)			
Возможные StatusCode	0, -999, -10000, -1601			
Пример	https://mtokentest.kaspi.kz:8543/r1/v01/payment/status/35134863 Ответ: { "StatusCode": 0, "Message": "OK", "Data": { "Status": "Wait" } }			

### 2.4. Возврат покупки

Как сделать возврат покупки?

Возврат осуществляется от лица кассира в приложении Kaspi Pay в рамках одной торговой точки. В приложении Kaspi Pay перейдите в раздел «Возврат».

- попросите покупателя отсканировать QR-код с экрана мобильного POS;
- выберите нужную покупку и нажмите «Оформить возврат»;
- введите сумму возврата и подтвердите операцию.

Деньги поступят покупателю моментально.

## Вторая схема Стандартный вариант доступа

### 3. Вторая схема Стандартный вариант доступа

Я хочу интегрироваться по стандартной схеме, что мне для этого нужно?

- 1) Зарегистрироваться в приложении Kaspi Pay – [ссылка](#).
- 2) Настроить точки продаж и кассиров – [ссылка](#).
- 3) Отправить электронной почтой запрос в банк, для выпуска сертификата, см Приложение 1.
- 4) Использовать полученный сертификат при обращении к методам API.

#### 3.1. Формат данных API

- Формат передачи данных: JSON (Content-Type: application/json)
- Формат даты: ISO 8601 (YYYY-MM-DDThh:mm:ss±h). Пример: 2020-08-09T18:31:42+06
- В каждый запрос к API нужно добавлять HTTP-заголовок “X-Request-ID” с уникальным значением (GUID/UID), генерируемый клиентом самостоятельно. Пример: X-Request-ID: da169655f65740d991b1bbfcd9802b28
- Все ответы имеют базовую структуру:

```
{
  "StatusCode": 0,
  "Message": "OK",
  "Data": {...}
}
```

Наименование	Тип	Обязательный	Описание
StatusCode	Int	Да	Код результата
Message	String	Нет	Сообщение результата, «OK» или «текст ошибки»
Data	Object	Нет	Объект данных, если имеется

Список StatusCode и их значения можно получить в Приложении 3.

#### 3.2. API List

##### 3.2.1. Регистрация кассы/устройства партнера

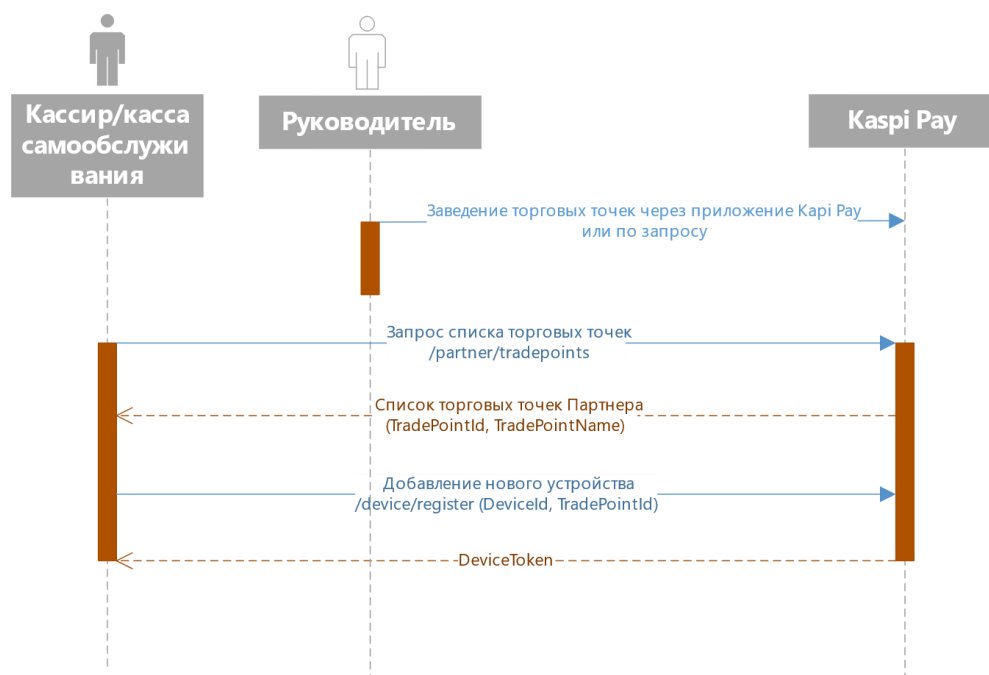


Рисунок 3.3 Регистрация устройства

## Вторая схема Стандартный вариант доступа

Перед регистрацией устройства нужно получить список торговых точек (см. п. 3.2.2) и выбрав нужную торговую точку вызвать метод добавления нового устройства (см. п. 3.2.3).

### 3.2.2. Список торговых точек

Описание	Получение списка торговых точек партнера для добавления устройства
Поставщик	Kaspi Pay
Потребитель	Партнер
Метод	GET
Адрес	https://mtokentest.kaspi.kz:8544/r2/v01/partner/tradepoints
Входные параметры	
Выходные параметры	TradePointId: Int64 – идентификатор торговой точки TradePointName: String – имя торговой точки
Возможные StatusCode	0, -999, -10000, -14000002
Пример	<p>https://mtokentest.kaspi.kz:8544/r2/v01/partner/tradepoints</p> <p>Ответ: {     "StatusCode": 0,     "Message": "OK",     "Data": [         { "TradePointId": 1, "TradePointName": "Магазин 1 (Ауэзова, 65)" },         { "TradePointId": 2, "TradePointName": "Магазин 2 (Жарокова, 48)" }     ] }</p>

### 3.2.3. Регистрация устройства

Описание	Регистрация нового устройства в Kaspi Pay. В ответ на повторный вызов с теми же аргументами возвращается тот же результат.			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8544/r2/v01/device/register			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceId	String	Да	Идентификатор устройства в системе Партнера (⚠ Допустимы только латинские буквы, цифры, тире и нижнее подчеркивание)
	TradePointId	Int64	Да	Идентификатор торговой точки партнера в Kaspi Pay (см. п. 3.2.2).
Выходные параметры	DeviceToken: String – идентификатор зарегистрированного в Kaspi Pay устройства			
Возможные StatusCode	0, -999, -10000, -1503, -99000002, 990000026			

## Вторая схема Стандартный вариант доступа

Пример	<pre>https://mtokentest.kaspi.kz:8544/r2/v01/device/register  Запрос: {   "DeviceId": "GFC-456397",   "TradePointId": 12 }  Ответ: {   "StatusCode": 0,   "Message": "OK",   "Data": {     "DeviceToken": "2be4cc91-5895-48f8-8bc2-86c7bd419b3b"   } }</pre>
--------	--

### 3.2.4. Удаление/отключение устройства

Описание	Удаление/отключения устройства			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8544/r2/v01/device/delete			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceToken	Int64	Да	Идентификатор добавленного в Kaspi Pay устройства (см. п. 3.2.3)
Выходные параметры	Нет			
Возможные StatusCode	0, -999, -10000, -1501			
Пример	<pre>POST https://mtokentest.kaspi.kz:8544/r2/v01/device/delete  Запрос: {   "DeviceToken": "2be4cc91-5895-48f8-8bc2-86c7bd419b3b" }  Ответ: {   "StatusCode": 0 }</pre>			



### 3.3. Оплата QR и по ссылке

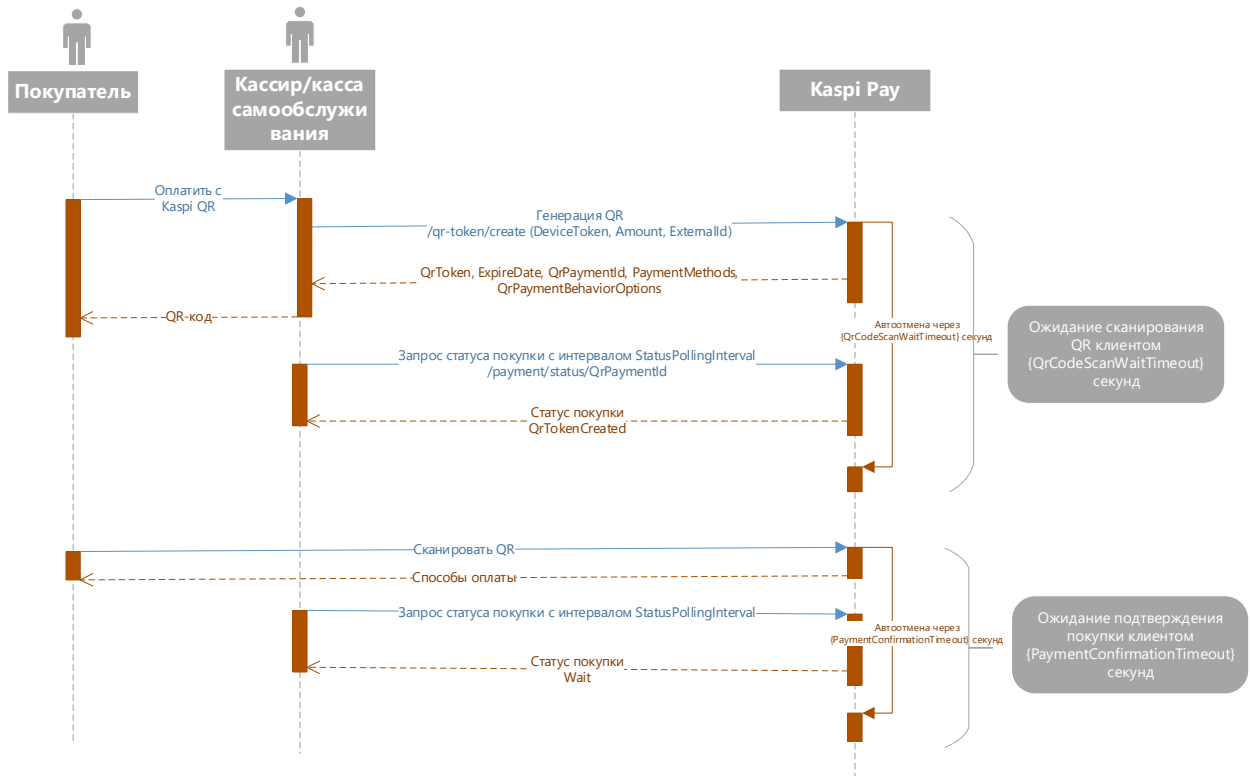


Рисунок 3.4 Процесс проведения покупки

Для того, чтобы начать процесс приема оплаты через QR-код, необходимо вызвать сервис создания QR-токена (см. п. 3.3.1). Вместе с QR-токеном в системе создается покупка. Полученный QR-токен нужно показать клиенту в виде изображения - QR-кода в приложении Партнера. Сразу после создания QR-токена, необходимо начать периодически запрашивать статус покупки (см. п. 3.3.3), с интервалом, указанным в выходном параметре сервиса создания QR-токена «StatusPollingInterval». Периодически запрашивать статус нужно до получения конечного статуса покупки, подробнее в Приложении 4. Также, QR-токен имеет ограниченное время жизни, которое состоит максимум из двух этапов:

**1) Ожидание сканирования клиентом.** Сразу после создания QR-токена, покупка имеет статус **QrTokenCreated**. Приложение Партнера должно зафиксировать «текущее время» + QrCodeScanWaitTimeout и если по истечении этого времени статус покупки не изменится и останется **QrTokenCreated**, то необходимо убрать отображение QR-кода и показать экран ошибки, т.к. QR-токен уже не действителен и сканирование QR-кода приведет к ошибке на стороне клиента.

Если зафиксированное время не истечет и статус покупки изменится на:

- **Wait**, то это означает, что клиент успешно отсканировал QR-код и желательно сменить экран отображения QR-кода на экран «ожидание подтверждения покупки» и перейти к следующему этапу.

- **Error**, то необходимо показать экран ошибки.

**2) Ожидание выбора счета оплаты и подтверждения покупки клиентом.** После успешного сканирования QR-кода клиентом, покупка имеет статус **Wait**. Приложение Партнера должно зафиксировать «текущее время» + PaymentConfirmationTimeout и если по истечении этого времени статус покупки не изменится и останется **Wait**, то необходимо сменить экран «ожидание подтверждения покупки» на экран ошибки. Покупка автоматически отменится и деньги вернутся клиенту. Если зафиксированное время не истечет и статус покупки изменится на:

## Вторая схема Стандартный вариант доступа

- **Processed**, то показать экран успешной покупки.
- **Error**, то необходимо показать экран ошибки.

### 3.3.1. Запрос на создание QR

При вызове метода создания QR-токена в ответе приходят следующие параметры, которые необходимо использовать при реализации приложения:

1. **PaymentMethods** – список способов оплаты, доступных Партнеру. [“Gold”, “Red”, “Loan”]  
Использовать при отображении с QR доступных способов оплаты.
2. **QrPaymentBehaviorOptions** – настройки временных интервалов для процессов QR-оплаты, использовать в логике работы приложения: получения статусов, смены экранов, вывода ошибок:
  - 2.1. **StatusPollingInterval** – интервал в секундах, через который необходимо периодически вызывать метод получения статуса операции
  - 2.2. **QrCodeScanWaitTimeout** – таймаут (в секундах) сканирования QR-кода (до получения статуса Wait).
  - 2.3. **PaymentConfirmationTimeout** – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)

Описание	Создание QR-токена			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8544/r2/v01/qr/create			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceToken	String	Да	Идентификатор добавленного в Kaspi Pay устройства
	Amount	Double	Да	Сумма покупки
	ExternalId	String	Нет	Идентификатор покупки в системе Партнера
Выходные параметры	QrToken: String – QR-токен ExpireDate: DateTime – дата и время истечения срока действия QR-токена. Сканирование QR-токена с истекшим сроком действия приводит к ошибке. QrPaymentId: Int64 – идентификатор покупки: используется для получения статуса покупки, возврата покупки. PaymentMethods: String[] – список способов оплаты, доступных Партнеру QrPaymentBehaviorOptions – настройки временных интервалов для процессов QR-оплаты: StatusPollingInterval – интервал в секундах, через который необходимо периодически вызывать статус операции (см. 3.3.3) QrCodeScanWaitTimeout – таймаут (в секундах) сканирования QR-кода (до получения статуса Wait). PaymentConfirmationTimeout – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)			
Возможные StatusCode	0, -999, -10000, -1501, -1502, -990000002, 9900000018, 9900000026, 9900000028, 9900000033			
Пример	https://mtokentest.kaspi.kz:8544/r2/v01/qr/create Запрос: { “DeviceToken”: “2be4cc91-5895-48f8-8bc2-86c7bd419b3b”, “Amount”: 200.00, “ExternalId”: “15” }			

## Вторая схема Стандартный вариант доступа

	<pre>}  Ответ: {   "StatusCode": 0,   "Message": "OK",   "Data": {     "QrToken": "51236903777280167836178166503744993984459",     "ExpireDate": "2020-01-16T10:30:00+06",     "QrPaymentId": 15,     "PaymentMethods": [ "Gold", "Red", "Loan"],     "QrPaymentBehaviorOptions": {       "StatusPollingInterval": 5,       "QrCodeScanWaitTimeout": 180,       "PaymentConfirmationTimeout": 65     }   } }</pre>
--	--

### 3.3.2. Запрос на создание ссылки на оплату

Данный сервис используется для приема оплаты через ссылку. В мобильном приложении партнера необходимо вызвать сервис создания ссылки на оплату (3.3.2). Полученную ссылку нужно использовать в мобильном приложении Партнера. После того, как клиент активирует ссылку (перейдет по ссылке) в мобильном приложении партнера будет произведено переправление в **мобильное приложение Kaspi.kz** (должно быть установлено) на стороне клиента. Если мобильное приложение Kaspi.kz не установлено у клиента, то активация ссылки приведет к открытию главной страницы сайта Kaspi.kz в мобильном браузере.

Сразу после создания ссылки, необходимо начать периодически запрашивать статус покупки (см. п. 3.3.3), с интервалом, указанным в выходном параметре сервиса создания ссылки «StatusPollingInterval». Периодически запрашивать статус нужно до получения конечного статуса покупки, подробнее в Приложении 4. Также, ссылка имеет ограниченное время жизни, которое состоит максимум из двух этапов:

- 1) **Ожидание активации ссылки клиентом.** Сразу после создания ссылки, покупка имеет статус **QrTokenCreated**. Приложение Партнера должно зафиксировать «текущее время» + QrCodeScanWaitTimeout и если по истечении этого времени статус покупки не изменится и останется **QrTokenCreated**, то необходимо убрать элементы связанные с ссылкой, т.к. ссылка уже не действительна и переход по ней приведет к ошибке на стороне клиента.

Если зафиксированное время не истечет и статус покупки изменится на:

- **Wait**, то это означает, что клиент успешно активировал ссылку (перешел по ссылке).
- **Error**, то необходимо показать экран ошибки.

- 2) **Ожидание выбора счета оплаты и подтверждения покупки клиентом.** После успешной активации ссылки клиентом, покупка имеет статус **Wait**. Приложение Партнера должно зафиксировать «текущее время» + PaymentConfirmationTimeout и если по истечении этого времени статус покупки не изменится и останется **Wait**, то необходимо показать экран ошибки. Покупка автоматически отменится и деньги вернутся клиенту. Если зафиксированное время не истечет и статус покупки изменится на:

- **Processed**, то показать экран успешной покупки.
- **Error**, то необходимо показать экран ошибки.

## Вторая схема Стандартный вариант доступа

При вызове метода создания ссылки на оплату в ответе приходят следующие параметры, которые необходимо использовать при реализации приложения:

1. **PaymentMethods** – список способов оплаты, доступных Партнеру. [“Gold”, “Red”, “Loan”]
2. **PaymentBehaviorOptions** – настройки временных интервалов для процессов оплаты, использовать в логике работы приложения: получения статусов, смены экранов, вывода ошибок:
  - 2.1. **StatusPollingInterval** – интервал в секундах, через который необходимо периодически вызывать метод получения статуса операции
  - 2.2. **LinkActivationWaitTimeout** – таймаут (в секундах) активации ссылки (до получения статуса Wait).
  - 2.3. **PaymentConfirmationTimeout** – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)

Описание	Создание ссылки на оплату			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8544/r2/v01/qr/create-link			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceToken	String	Да	Идентификатор добавленного в Kaspi Pay устройства
	Amount	Double	Да	Сумма покупки
	ExternalId	String	Нет	Идентификатор покупки в системе Партнера
Выходные параметры	PaymentLink: String – ссылка на оплату ExpireDate: DateTime – дата и время истечения срока действия ссылки. Переход по ссылке с истекшим сроком действия приводит к ошибке. PaymentId: Int64 – идентификатор покупки: используется для получения статуса покупки, возврата покупки. PaymentMethods: String[] – список способов оплаты, доступных Партнеру PaymentBehaviorOptions – настройки временных интервалов для процессов оплаты: StatusPollingInterval – интервал в секундах, через который необходимо периодически вызывать статус операции (см.п. 3.3.3) LinkActivationWaitTimeout – таймаут (в секундах) активации ссылки (до получения статуса Wait). PaymentConfirmationTimeout – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)			
Возможные StatusCode	0, -999, -10000, -1501, -1502, - 990000002, 9900000018, 9900000026, 9900000028, 9900000033			
Пример	https://mtokentest.kaspi.kz:8544/r2/v01/qr/create-link Запрос: { “DeviceToken”: “2be4cc91-5895-48f8-8bc2-86c7bd419b3b”, “Amount”: 200.00, “ExternalId”: “15” }  Ответ: { “StatusCode”: 0,			

## Вторая схема Стандартный вариант доступа

	<pre>"Message": "OK", "Data": {   "PaymentLink":     "https://pay.kaspi.kz/pay/59217969256592032202439955771435979527     677",   "ExpireDate": "2020-01-16T10:30:00+06",   "PaymentId": 15,   "PaymentMethods": [ "Gold", "Red", "Loan" ],   "PaymentBehaviorOptions": {     "StatusPollingInterval": 5,     "LinkActivationWaitTimeout": 180,     "PaymentConfirmationTimeout": 65   } }</pre>
--	--

### 3.3.3. Получение статуса операции

Описание	<p>Получение текущего статуса покупки. Список возможных статусов смотрите в Приложении 4. Статус операции необходимо запрашивать с периодичностью каждые n-секунд, указанной в значении <code>StatusPollingInterval</code>, полученном при генерации. Статус покупки необходимо запрашивать до тех пор, пока не получен конечный статус, или не истекло время жизни токена.</p> <p>Нормальный жизненный цикл покупки: QrTokenCreated -&gt; Wait -&gt; Processed -&gt; Error</p>			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	GET			
Адрес	https://mtokentest.kaspi.kz:8544/r2/v01/payment/status/{QrPaymentId}			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	QrPaymentId	Int64	Да	Идентификатор покупки
Выходные параметры	Status: String – статус операции (см. Приложение 4)			
Возможные StatusCode	0, -999, -10000, -1601			
Пример	<p>https://mtokentest.kaspi.kz:8544/r2/v01/payment/status/35134863</p> <p>Ответ:</p> <pre>{   "StatusCode": 0,   "Message": "OK",   "Data": {     "Status": "Wait"   } }</pre>			

## Вторая схема Стандартный вариант доступа

### 3.4. Возврат покупки с участием покупателя

Для того, чтобы начать процесс возврата покупки через QR-код, необходимо вызвать сервис создания QR-токена возврата (см. п. 3.4.1). Вместе с QR-токеном в системе создается операция возврата. Полученный QR-токен нужно показать клиенту в виде изображения QR-кода в приложении Партнера. Сразу после создания QR-токена, необходимо начать периодически запрашивать статус возврата (см. п.3.4.2), с интервалом, указанным в выходном параметре сервиса создания QR-токена «QrCodeScanEventPollingInterval». Периодически запрашивать статус нужно до смены статуса возврата с QrTokenCreated -> CustomerIdentityObtained. После получения статуса возврата CustomerIdentityObtained, нужно вызвать сервис получения списка покупок клиента (см. п. 3.4.3). В полученном списке покупок выбирается нужная покупка и вызывается сервис получения деталей операции (см. п. 3.4.4). В ответе сервиса придет значение AvailableReturnAmount, которое необходимо учитывать при оформлении возврата. Далее вызывается метод возврата покупки (см. п. 3.4.5).

#### 3.4.1. Создание QR-токена возврата

При вызове метода создания QR-токена в ответе приходят следующие параметры, которые необходимо использовать при реализации приложения:

**QrReturnBehaviorOptions** – настройки временных интервалов для процессов QR-возврата, использовать в логике работы приложения: получения статусов, смены экранов, вывода ошибок:

1. **QrCodeScanEventPollingInterval** – интервал в секундах, через который необходимо периодически вызывать метод получения статуса операции
2. **QrCodeScanWaitTimeout** – таймаут (в секундах) сканирования QR-кода (до получения статуса CustomerIdentityObtained).

Описание	Создание QR-токена возврата			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8544/r2/v01/return/create			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceToken	String	Да	Идентификатор добавленного в Kaspi Pay устройства
	ExternalId	String	Нет	Идентификатор возврата в системе Партнера
Выходные параметры	QrToken: String – QR-токен ExpireDate: DateTime – дата и время истечения срока действия QR-токена. Сканирование QR-токена с истекшим сроком действия приводит к ошибке. QrReturnId: Int64 – идентификатор возврата: используется для получения статуса возврата. QrReturnBehaviorOptions – настройки временных интервалов для процессов QR-оплаты: QrCodeScanEventPollingInterval – интервал в секундах, через который необходимо периодически вызывать статус операции (см. п. 3.4.2) QrCodeScanWaitTimeout – таймаут (в секундах) сканирования QR-кода (до получения статуса CustomerIdentityObtained).			
Возможные StatusCode	0, -999, -10000, -1501, -1502, -990000002, 9900000018			
Пример	https://mtokentest.kaspi.kz:8544/r2/v01/return/create Запрос: {			

## Вторая схема Стандартный вариант доступа

	<pre>"DeviceToken": "2be4cc91-5895-48f8-8bc2-86c7bd419b3b", "ExternalId": "15" }  Ответ: {   "StatusCode": 0,   "Message": "OK",   "Data": {     "QrToken": "51236903777280167836178166503744993984459",     "ExpireDate": "2020-01-16T10:30:00+06",     "QrReturnId": 15,     "QrReturnBehaviorOptions": {       "QrCodeScanEventPollingInterval": 5,       "QrCodeScanWaitTimeout": 180     }   } }</pre>
--	---

### 3.4.2. Получение текущего статуса возврата

Описание	<p>Получение текущего статуса возврата. Список возможных статусов смотрите в Приложении 5.</p> <p>Статус операции необходимо запрашивать с периодичностью каждые n-секунд, указанной в значении QrCodeScanEventPollingInterval, полученном при генерации QR-токена возврата. Статус возврата необходимо запрашивать до тех пор, пока не получен конечный статус, или не истекло время жизни токена.</p> <p>Нормальный жизненный цикл возврата: QrTokenCreated -&gt; CustomerIdentityObtained-&gt; Error</p>			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	GET			
Адрес	https://mtokentest.kaspi.kz:8544/r2/v01/return/status/{QrReturnId}			
Входные параметры	Наименование	Тип	Обязательный	Описание
	QrReturnId	Int64	Да	Идентификатор возврата
Выходные параметры	Status: String – статус операции (см. Приложение 5)			
Возможные StatusCode	0, -999, -10000, -1601			
Пример	<pre>https://mtokentest.kaspi.kz:8544/r2/v01/return/status/35134863  Ответ: {   "StatusCode": 0,   "Message": "OK",   "Data": {     "Status": "QrTokenCreated"   } }</pre>			

## Вторая схема Стандартный вариант доступа

### 3.4.3. Получение списка покупок клиента

Описание	Получение списка последних N покупок клиента (N <=10 )			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8544/r2/v01/return/operations			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceToken	String	Да	Идентификатор добавленного в Kaspi Pay устройства
	QrReturnId	Int64	Да	Идентификатор возврата
	MaxResult	Int64	Нет	Число покупок клиента (по умолчанию 10). Максимально возможное значение <b>10</b> .
Выходные параметры	QrPaymentId: Int64 – идентификатор операции Amount: Double – общая сумма покупки TransactionDate: DateTime – дата и время совершения покупки			
Возможные StatusCode	0, -999, -10000, -1601			
Пример	<p>https://mtokentest.kaspi.kz:8544/r2/v01/return/operations</p> <p>Ответ:</p> <pre>{   "StatusCode": 0,   "Message": "OK",   "Data":     [       {         "QrPaymentId": 900077110,         "TransactionDate": "2021-12-08T11:35:48.1566667",         "Amount": 1.00       }     ] }</pre>			

### 3.4.4. Детали операции

Описание	Возвращает детали покупки по идентификатору. Может быть использован в процессе возврата для верификации покупки.			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	GET			
Адрес	https://mtokentest.kaspi.kz:8544/r2/v01/payment/details			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	QrPaymentId	Int64	Да	Идентификатор покупки
	DeviceToken	String	Да	Идентификатор устройства в Kaspi Pay
Выходные параметры	QrPaymentId: Int64 – идентификатор операции TotalAmount: Double – общая сумма покупки AvailableReturnAmount: Double – сумма, доступная к возврату TransactionDate: DateTime – дата и время совершения покупки			



**Вторая схема** Стандартный вариант доступа

Возможные StatusCode	0, -999, -10000, -1601
Пример	<p><a href="https://mtokentest.kaspi.kz:8544/r2/v01/payment/details?QrPaymentId=123&amp;DeviceToken=2be4cc91-5895-48f8-8bc2-86c7bd419b3b">https://mtokentest.kaspi.kz:8544/r2/v01/payment/details?QrPaymentId=123&amp;DeviceToken=2be4cc91-5895-48f8-8bc2-86c7bd419b3b</a></p> <p>Ответ:</p> <pre>{   "Data": {     "QrPaymentId": 501020796,     "TotalAmount": 11.00,     "AvailableReturnAmount": 11.00,     "TransactionDate": "2021-11-3T11:55:14.166+06:00"   },   "StatusCode": 0 }</pre>

**3.4.5. Частичный или полный возврат покупки**

Описание	Частичный или полный возврат покупки. Возврат покупки возможен только в рамках одной торговой точки Kaspi Pay. Возврат можно выполнить с другого устройства, смарт-поса или от лица кассира в приложении Kaspi Pay в рамках одной торговой точки.			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	<a href="https://mtokentest.kaspi.kz:8544/r2/v01/payment/return">https://mtokentest.kaspi.kz:8544/r2/v01/payment/return</a>			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceToken	String	Да	Идентификатор устройства в Kaspi Pay
	QrPaymentId	Int64	Да	Идентификатор покупки
	QrReturnId	Int64	Да	Идентификатор возврата
	Amount	Double	Да	Сумма возврата
Выходные параметры	ReturnOperationId: Int64 – идентификатор возврата			
Возможные StatusCode	0, -999, -10000, -1501, -1502, -1601, -990000005, -990000006, -990000001, -990000002, -990000003, -990000011, -990000020, 990000018			
Пример	<p><a href="https://mtokentest.kaspi.kz:8544/r2/v01/payment/return">https://mtokentest.kaspi.kz:8544/r2/v01/payment/return</a></p> <p>Запрос:</p> <pre>{   "DeviceToken": "2be4cc91-5895-48f8-8bc2-86c7bd419b3b",   "QrPaymentId": 123,   "QrReturnId": 13,    "Amount": 200.00 }</pre> <p>Ответ:</p> <pre>{   "StatusCode": 0,   "Message": "OK",   "Data": {     "ReturnOperationId": 15   } }</pre>			

### Третья схема Вариант с усиленной безопасностью

#### 4. Третья схема Вариант с усиленной безопасностью

Я хочу интегрироваться по схеме с усиленной безопасностью, что мне для этого нужно?

- 1) Организации партнера должны зарегистрироваться в приложении Kaspi Pay – [ссылка](#).
- 2) Для каждой организации настроить точки продаж – [ссылка](#).
- 3) Отправить электронной почтой запрос в банк, для выпуска сертификата, см Приложение 1.
- 4) Сообщить банку статический IP-адрес сервера, с которого будут отправляться запросы к API.
- 5) Предоставить в банк информацию об организациях в пользу которых будет проводиться прием платежей (БИН/ИИН).
- 6) Использовать полученный сертификат при обращении к методам API.

#### 4.1. Формат данных API

- Формат передачи данных: JSON (Content-Type: application/json)
- Формат даты: ISO 8601 (YYYY-MM-DDThh:mm:ss±h). Пример: 2020-08-09T18:31:42+06
- В каждый запрос к API нужно добавлять HTTP-заголовок "X-Request-ID" с уникальным значением (GUID/UID), генерируемый клиентом самостоятельно. Пример: X-Request-ID: da169655f65740d991b1bbfcd9802b28
- Все ответы имеют базовую структуру:

```
{
  "StatusCode": 0,
  "Message": "OK",
  "Data": {...}
}
```

Наименование	Тип	Обязательный	Описание
StatusCode	Int	Да	Код результата
Message	String	Нет	Сообщение результата, «OK» или «текст ошибки»
Data	Object	Нет	Объект данных, если имеется

Список StatusCode и их значения можно получить в Приложении 3

Третья схема Вариант с усиленной безопасностью

4.2.API List

4.2.1. Регистрация кассы/устройства партнера

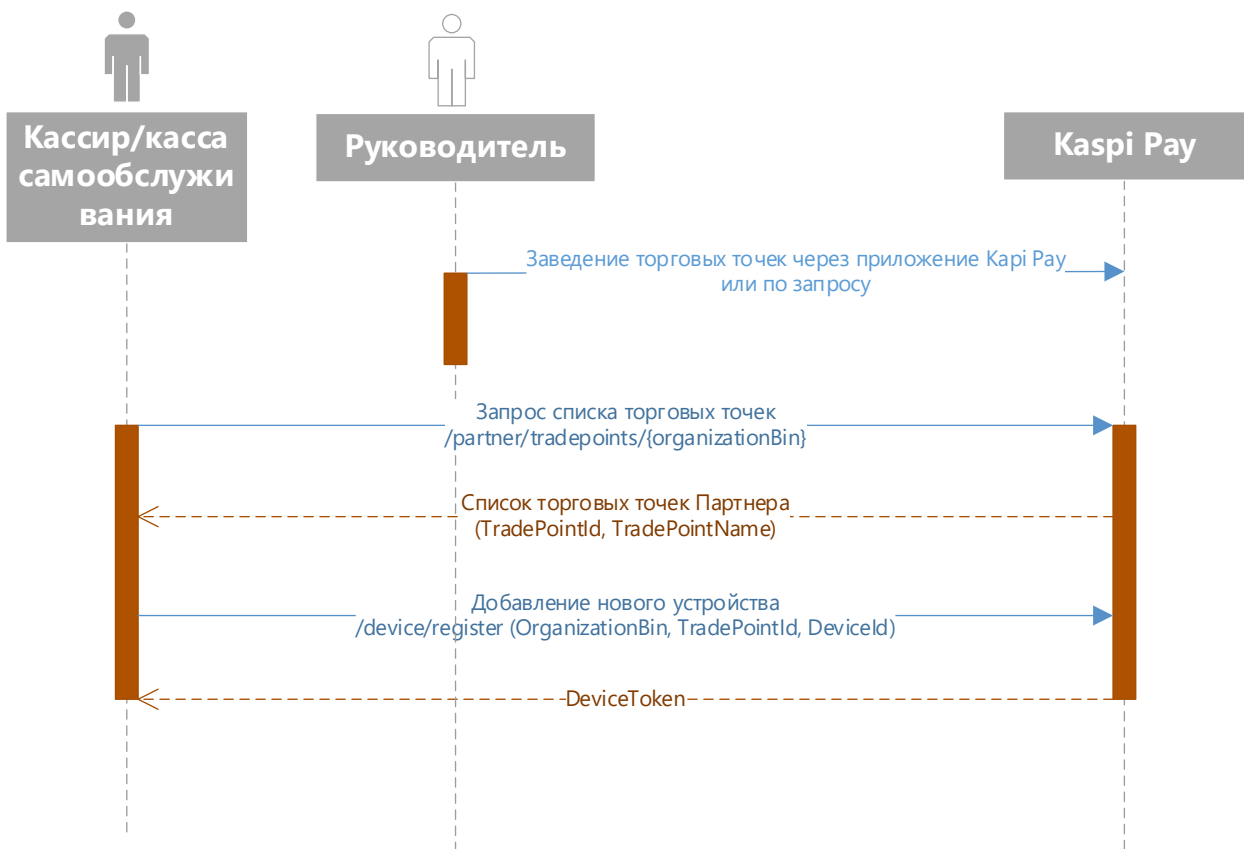


Рисунок 4.5 Регистрация устройства

Перед регистрацией устройства нужно получить список торговых точек (см. п. 4.2.2) и выбрав нужную торговую точку вызвать метод добавления нового устройства (см. п. 4.2.3).

4.2.2. Список торговых точек

Описание	Получение списка торговых точек партнера для добавления устройства			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	GET			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/partner/tradepoints/{organizationBin}			
Входные параметры	Наименование	Тип	Обязательный	Описание
	organizationBin	String	Да	БИН/ИИН организации
Выходные параметры	TradePointId: Int64 – идентификатор торговой точки TradePointName: String – имя торговой точки			
Возможные StatusCode	0, -999, -10000, -14000002			
Пример	https://mtokentest.kaspi.kz:8545/r3/v01/partner/tradepoints/180340021791  Ответ:{ "StatusCode": 0, "Message": "OK", "Data": [ 			

**Третья схема** Вариант с усиленной безопасностью

	<pre> { "TradePointId": 1, "TradePointName": "Магазин 1 (Ауэзова, 65)" }, { "TradePointId": 2, "TradePointName": "Магазин 2 (Жарокова, 48)" } ] } </pre>
--	--

**4.2.3. Регистрация устройства**

Описание	Регистрация нового устройства в Kaspi Pay. В ответ на повторный вызов с теми же аргументами возвращается тот же результат.			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/device/register			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceId	String	Да	Идентификатор устройства в системе Партнера (⚠ Допустимы только латинские буквы, цифры, тире и нижнее подчеркивание)
	TradePointId	Int64	Да	Идентификатор торговой точки партнера в Kaspi Pay (см. п. 4.2.2).
	OrganizationBin	String	Да	БИН/ИИН организации
Выходные параметры	DeviceToken: String – идентификатор зарегистрированного в Kaspi Pay устройства			
Возможные StatusCode	0, -999, -10000, -1503, -99000002, 990000026			
Пример	<p>https://mtokentest.kaspi.kz:8545/r3/v01/device/register</p> <p>Запрос:</p> <pre> {   "OrganizationBin": "180340021791",   "DeviceId": "GFC-456397",   "TradePointId": 12 } </pre> <p>Ответ:</p> <pre> {   "StatusCode": 0,   "Message": "OK",   "Data": {     "DeviceToken": "2be4cc91-5895-48f8-8bc2-86c7bd419b3b"   } } </pre>			

**4.2.4. Удаление/отключение устройства**

Описание	Удаление/отключения устройства			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/device/delete			
	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>

**Третья схема** Вариант с усиленной безопасностью

Входные параметры	DeviceToken	Int64	Да	Идентификатор добавленного в Kaspi Pay устройства (см. п. 4.2.3)
	OrganizationBin	String	Да	БИН/ИИН организации
Выходные параметры	Нет			
Возможные StatusCode	0, -999, -10000, -1501			
Пример	<p>POST https://mtokentest.kaspi.kz:8545/r3/v01/device/delete</p> <p>Запрос:</p> <pre>{   "OrganizationBin": "180340021791",   "DeviceToken": "2be4cc91-5895-48f8-8bc2-86c7bd419b3b" }</pre> <p>Ответ:</p> <pre>{   "StatusCode": 0 }</pre>			

## 4.3. Оплата QR и по ссылке

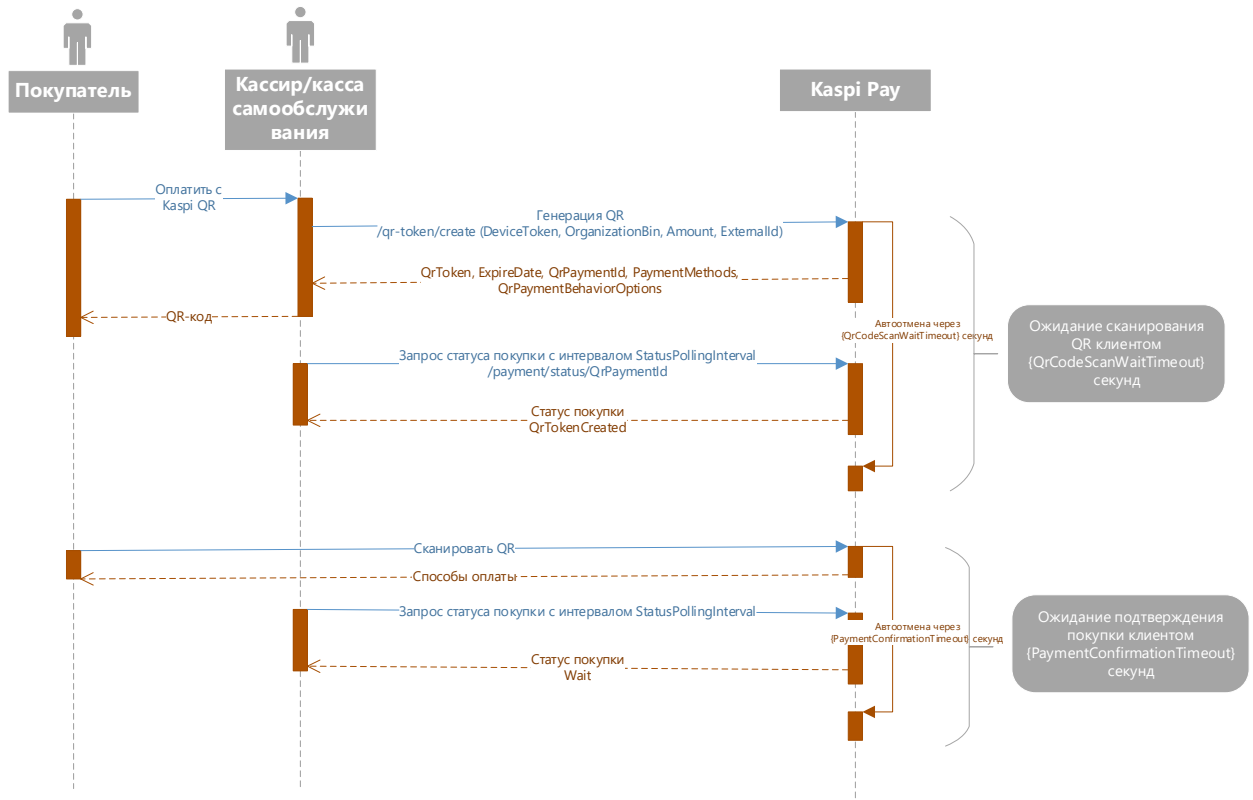


Рисунок 6. Процесс проведения покупки

Для того, чтобы начать процесс приема оплаты через QR-код, необходимо вызвать сервис создания QR-токена (см. п. 4.3.1). Вместе с QR-токеном в системе создается покупка. Полученный QR-токен нужно показать клиенту в виде изображения - QR-кода в приложении Партнера. Сразу после создания QR-токена, необходимо начать периодически запрашивать статус покупки (см. п. 4.3.3), с интервалом, указанным в выходном параметре сервиса создания QR-токена «StatusPollingInterval». Периодически запрашивать статус нужно до получения конечного статуса покупки, подробнее в Приложении 4. Также, QR-токен имеет ограниченное время жизни, которое состоит максимум из двух этапов:

1) **Ожидание сканирования клиентом.** Сразу после создания QR-токена, покупка имеет статус **QrTokenCreated**. Приложение Партнера должно зафиксировать «текущее время» + QrCodeScanWaitTimeout и если по истечении этого времени статус покупки не изменится и останется **QrTokenCreated**, то необходимо убрать отображение QR-кода и показать экран ошибки, т.к. QR-токен уже не действителен и сканирование QR-кода приведет к ошибке на стороне клиента.

Если зафиксированное время не истечет и статус покупки изменится на:

- **Wait**, то это означает, что клиент успешно отсканировал QR-код и желательно сменить экран отображения QR-кода на экран «ожидание подтверждения покупки» и перейти к следующему этапу.

- **Error**, то необходимо показать экран ошибки.

2) **Ожидание выбора счета оплаты и подтверждения покупки клиентом.** После успешного сканирования QR-кода клиентом, покупка имеет статус **Wait**. Приложение Партнера должно зафиксировать «текущее время» + PaymentConfirmationTimeout и если по истечении этого времени статус покупки не изменится и останется **Wait**, то необходимо сменить экран «ожидание подтверждения покупки» на экран ошибки. Покупка автоматически отменится и деньги вернутся клиенту. Если зафиксированное время не истечет и статус покупки изменится на:

### Третья схема Вариант с усиленной безопасностью

- **Processed**, то показать экран успешной покупки.
- **Error**, то необходимо показать экран ошибки.

#### 4.3.1. Запрос на создание QR

При вызове метода создания QR-токена в ответе приходят следующие параметры, которые необходимо использовать при реализации приложения:

1. **PaymentMethods** – список способов оплаты, доступных Партнеру. [“Gold”, “Red”, “Loan”]  
Использовать при отображении с QR доступных способов оплаты.
2. **QrPaymentBehaviorOptions** – настройки временных интервалов для процессов QR-оплаты, использовать в логике работы приложения: получения статусов, смены экранов, вывода ошибок:
  - 2.1. **StatusPollingInterval** – интервал в секундах, через который необходимо периодически вызывать метод получения статуса операции
  - 2.2. **QrCodeScanWaitTimeout** – таймаут (в секундах) сканирования QR-кода (до получения статуса Wait).
  - 2.3. **PaymentConfirmationTimeout** – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)

Описание	Создание QR-токена			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/qr/create			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	OrganizationBin	String	Да	БИН/ИИН организации
	DeviceToken	String	Да	Идентификатор добавленного в Kaspi Pay устройства
	Amount	Double	Да	Сумма покупки
	ExternalId	String	Нет	Идентификатор покупки в системе Партнера
Выходные параметры	QrToken: String – QR-токен ExpireDate: DateTime – дата и время истечения срока действия QR-токена. Сканирование QR-токена с истекшим сроком действия приводит к ошибке. QrPaymentId: Int64 – идентификатор покупки: используется для получения статуса покупки, возврата покупки. PaymentMethods: String[] – список способов оплаты, доступных Партнеру QrPaymentBehaviorOptions – настройки временных интервалов для процессов QR-оплаты: StatusPollingInterval – интервал в секундах, через который необходимо периодически вызывать статус операции (см. 4.3.3) QrCodeScanWaitTimeout – таймаут (в секундах) сканирования QR-кода (до получения статуса Wait). PaymentConfirmationTimeout – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)			
Возможные StatusCode	0, -999, -10000, -1501, -1502, -990000002, 9900000018, 9900000026, 9900000028, 9900000033			
Пример	https://mtokentest.kaspi.kz:8545/r3/v01/qr/create Запрос: { “OrganizationBin”: “180340021791”, “DeviceToken”: “2be4cc91-5895-48f8-8bc2-86c7bd419b3b”,			

### Третья схема Вариант с усиленной безопасностью

	<pre>    "Amount": 200.00,     "ExternalId": "15"   }   Ответ:   {     "StatusCode": 0,     "Message": "OK",     "Data": {       "QrToken": "51236903777280167836178166503744993984459",       "ExpireDate": "2020-01-16T10:30:00+06",       "QrPaymentId": 15,       "PaymentMethods": [ "Gold", "Red", "Loan" ],       "QrPaymentBehaviorOptions": {         "StatusPollingInterval": 5,         "QrCodeScanWaitTimeout": 180,         "PaymentConfirmationTimeout": 65       }     }   } }</pre>
--	---

#### 4.3.2. Создание ссылки на оплату

Данный сервис используется для приема оплаты через ссылку. В мобильном приложении партнера необходимо вызвать сервис создания ссылки на оплату (4.3.2). Полученную ссылку нужно использовать в мобильном приложении Партнера. После того, как клиент активирует ссылку (перейдет по ссылке) в мобильном приложении партнера будет произведено переправление в **мобильное приложение Kaspi.kz** (должно быть установлено) на стороне клиента. Если мобильное приложение Kaspi.kz не установлено у клиента, то активация ссылки приведет к открытию главной страницы сайта Kaspi.kz в мобильном браузере.

Сразу после создания ссылки, необходимо начать периодически запрашивать статус покупки (см. п. 4.3.3), с интервалом, указанным в выходном параметре сервиса создания ссылки «StatusPollingInterval». Периодически запрашивать статус нужно до получения конечного статуса покупки, подробнее в Приложении 4. Также, ссылка имеет ограниченное время жизни, которое состоит максимум из двух этапов:

- 1) **Ожидание активации ссылки клиентом.** Сразу после создания ссылки, покупка имеет статус **QrTokenCreated**. Приложение Партнера должно зафиксировать «текущее время» + QrCodeScanWaitTimeout и если по истечении этого времени статус покупки не изменится и останется **QrTokenCreated**, то необходимо убрать элементы связанные с ссылкой, т.к. ссылка уже не действительна и переход по ней приведет к ошибке на стороне клиента.

Если зафиксированное время не истечет и статус покупки изменится на:

- **Wait**, то это означает, что клиент успешно активировал ссылку (перешел по ссылке).
- **Error**, то необходимо показать экран ошибки.

- 2) **Ожидание выбора счета оплаты и подтверждения покупки клиентом.** После успешной активации ссылки клиентом, покупка имеет статус **Wait**. Приложение Партнера должно зафиксировать «текущее время» + PaymentConfirmationTimeout и если по истечении этого времени статус покупки не изменится и останется **Wait**, то необходимо показать экран ошибки. Покупка автоматически отменится и деньги вернутся клиенту. Если зафиксированное время не истечет и статус покупки изменится на:

- **Processed**, то показать экран успешной покупки.
- **Error**, то необходимо показать экран ошибки.



### Третья схема Вариант с усиленной безопасностью

При вызове метода создания ссылки на оплату в ответе приходят следующие параметры, которые необходимо использовать при реализации приложения:

1. **PaymentMethods** – список способов оплаты, доступных Партнеру. [“Gold”, “Red”, “Loan”]
2. **PaymentBehaviorOptions** – настройки временных интервалов для процессов оплаты, использовать в логике работы приложения: получения статусов, смены экранов, вывода ошибок:
  - 2.1. **StatusPollingInterval** – интервал в секундах, через который необходимо периодически вызывать метод получения статуса операции
  - 2.2. **LinkActivationWaitTimeout** – таймаут (в секундах) активации ссылки (до получения статуса Wait).
  - 2.3. **PaymentConfirmationTimeout** – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)

Описание	Создание ссылки на оплату			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/qr/create-link			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	OrganizationBin	String	Да	БИН/ИИН организации
	DeviceToken	String	Да	Идентификатор добавленного в Kaspi Pay устройства
	Amount	Double	Да	Сумма покупки
	ExternalId	String	Нет	Идентификатор покупки в системе Партнера
Выходные параметры	PaymentLink: String – ссылка на оплату ExpireDate: DateTime – дата и время истечения срока действия ссылки. Переход по ссылке с истекшим сроком действия приводит к ошибке. PaymentId: Int64 – идентификатор покупки: используется для получения статуса покупки, возврата покупки. PaymentMethods: String[] – список способов оплаты, доступных Партнеру PaymentBehaviorOptions – настройки временных интервалов для процессов оплаты: StatusPollingInterval – интервал в секундах, через который необходимо периодически вызывать статус операции (см.п. 4.3.3) LinkActivationWaitTimeout – таймаут (в секундах) активации ссылки (до получения статуса Wait). PaymentConfirmationTimeout – таймаут (в секундах) ожидания подтверждения покупки клиентом (после получения статуса Wait)			
Возможные StatusCode	0, -999, -10000, -1501, -1502, - 990000002, 9900000018, 9900000026, 9900000028, 9900000033			
Пример	https://mtokentest.kaspi.kz:8545/r3/v01/qr/create-link Запрос: { “OrganizationBin”: “180340021791”, “DeviceToken”: “2be4cc91-5895-48f8-8bc2-86c7bd419b3b”, “Amount”: 200.00, “ExternalId”: “15” } Ответ:			

**Третья схема** Вариант с усиленной безопасностью

	<pre>{   "StatusCode": 0,   "Message": "OK",   "Data": {     "PaymentLink":     "https://pay.kaspi.kz/pay/59217969256592032202439955771435979527     677",     "ExpireDate": "2020-01-16T10:30:00+06",     "PaymentId": 15,     "PaymentMethods": [ "Gold", "Red", "Loan" ],     "PaymentBehaviorOptions": {       "StatusPollingInterval": 5,       "LinkActivationWaitTimeout": 180,       "PaymentConfirmationTimeout": 65     }   } }</pre>
--	---

#### 4.3.3. Получение статуса операции

Описание	<p>Получение текущего статуса покупки. Список возможных статусов смотрите в Приложении 4. Статус операции необходимо запрашивать с периодичностью каждые n-секунд, указанной в значении StatusPollingInterval, полученном при генерации QR (⚠ <b>удаленная оплата является исключением</b>). Статус покупки необходимо запрашивать до тех пор, пока не получен конечный статус, или не истекло время жизни токена.</p> <p>Нормальный жизненный цикл покупки: QrTokenCreated -&gt; Wait -&gt; Processed -&gt; Error</p>			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	GET			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/payment/status/{QrPaymentId}			
Входные параметры	Наименование	Тип	Обязательный	Описание
	QrPaymentId	Int64	Да	Идентификатор покупки
Выходные параметры	Status: String – статус операции (см. Приложение 4)			
Возможные StatusCode	0, -999, -10000, -1601			
Пример	<p>https://mtokentest.kaspi.kz:8545/r3/v01/payment/status/35134863</p> <p>Ответ:</p> <pre>{   "StatusCode": 0,   "Message": "OK",   "Data": {     "Status": "Wait"   } }</pre>			

## 4.4. Детали операции

Описание	Возвращает детали покупки по идентификатору. Может быть использован в процессе возврата для верификации покупки.			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	GET			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/payment/details			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	QrPaymentId	Int64	Да	Идентификатор покупки
	DeviceToken	String	Да	Идентификатор устройства в Kaspi Pay
Выходные параметры	QrPaymentId: Int64 – идентификатор операции TotalAmount: Double – общая сумма покупки AvailableReturnAmount: Double – сумма, доступная к возврату TransactionDate: DateTime – дата и время совершения покупки			
Возможные StatusCode	0, -999, -10000, -1601			
Пример	https://mtokentest.kaspi.kz:8545/r3/v01/payment/details?QrPaymentId=123&DeviceToken=2be4cc91-5895-48f8-8bc2-86c7bd419b3b  Ответ: <pre>{   "Data":   {     "QrPaymentId":501020796,     "TotalAmount":11.00,     "AvailableReturnAmount":11.00,     "TransactionDate":"2021-11-3T11:55:14.166+06:00"   },   "StatusCode":0 }</pre>			

## 4.5. Возврат покупки без участия покупателя

Описание	Частичный или полный возврат покупки. Возврат покупки возможен только в рамках одной торговой точки Kaspi Pay. Возврат можно выполнить с другого устройства, смарт-поса или от лица кассира в приложении Kaspi Pay в рамках одной торговой точки.			
Поставщик	Kaspi Pay			
Потребитель	Партнер			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/payment/return			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	DeviceToken	String	Да	Идентификатор устройства в Kaspi Pay
	QrPaymentId	Int64	Да	Идентификатор покупки
	Amount	Double	Да	Сумма возврата
	OrganizationBin	String	Да	БИН/ИИН организации
Выходные параметры	ReturnOperationId: Int64 – идентификатор возврата			

### Третья схема Вариант с усиленной безопасностью

Возможные StatusCode	0, -999, -10000, -1501, -1502, -1601, -990000005, -990000006, -990000001, -990000002, -990000003, -990000011, -990000020, 990000018
Пример	<p>https://mtokentest.kaspi.kz:8545/r3/v01/payment/return</p> <p>Запрос:</p> <pre>{   "DeviceToken": "2be4cc91-5895-48f8-8bc2-86c7bd419b3b",   "OrganizationBin": "180340021791",   "QrPaymentId": 123,   "Amount": 200.00 }</pre> <p>Ответ:</p> <pre>{   "StatusCode": 0,   "Message": "OK",   "Data": {     "ReturnOperationId": 15   } }</pre>

## 4.6. Удаленная оплата

Счет на оплату выставляется покупателю по номеру телефона. Для проверки номера покупателя необходимо вызвать сервис поиска получателя по номеру (см. п. 4.6.1). После проверки вызвать сервис выставления счета на удаленную оплату (см. п. 4.6.2). Покупатель получает счет, который может оплатить в течении 24-х часов. Для получения статуса оплаты нужно вызвать сервис получения статуса операции (см. п. 4.3.3). ⚠ Сервис получения статуса рекомендуется вызывать в ответ на какое-либо событие в системе партнера, например: нажатие кнопки «обновить статус покупки»; обновление страницы заказа и т.п. Запрещается вызывать сервис в автоматическом режиме через интервалы времени, т.к. система предотвращения ненормированных нагрузок на сервер отобьет автоматические запросы. Статус операции необходимо запрашивать до тех пор, пока не получен конечный статус.

### 4.6.1. Поиск получателя счета на оплату по номеру телефона

Описание	Поиск получателя счета на оплату по номеру телефона Данные сервис может вызываться на одном устройстве не чаще чем 50 раз в минуту.			
Поставщик	Kaspi Pay			
Потребитель	Онлайн-касса			
Метод	GET			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/remote/client-info			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	PhoneNumber	String	Да	Номер телефона клиента
	DeviceToken	Int64	Да	Идентификатор добавленного в Kaspi Pay устройства
Выходные параметры	ClientName: String - ФИО клиента			
Возможные StatusCode	0, -999, -1401, -1404, -1501, -990000002, 9900000018, 9900000029, 9900000031, 9900000032			
Пример	<p>https://mtokentest.kaspi.kz:8545/r3/v01/remote/client-info?phoneNumber=87077777777&amp;deviceToken=321</p> <p>Ответ:</p> <pre>{   "StatusCode": 0,</pre>			

**Третья схема** Вариант с усиленной безопасностью

	<pre> "Message": "OK", "Data": {   "ClientName": "ФИО" } } </pre>
--	---

**4.6.2. Выставление счета на удаленную оплату**

Описание	Выставление счета на удаленную оплату			
Поставщик	Kaspi Pay			
Потребитель	Онлайн-касса			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/remote/create			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	OrganizationBin	String	Да	БИН/ИИН организации
	Amount	Double	Да	Сумма к оплате
	PhoneNumber	String	Да	Номер телефона клиента
	DeviceToken	Int64	Да	Идентификатор добавленного в Kaspi Pay устройства
	Comment	String	Нет	Комментарий к оплате
Выходные параметры	QrPaymentId: Int64 - Идентификатор покупки			
Возможные StatusCode	0, -999, -1401, -1404, -1501, -99000002, 99000010, 990000018, 990000028, 990000032			
Пример	<p>https://mtokentest.kaspi.kz:8545/r3/v01/remote/create</p> <p>Запрос:</p> <pre> {   "Amount": 1.00,   "PhoneNumber": "8707777777",   "Comment": "Test",   "DeviceToken": 321,   "OrganizationBin": "180340021791" } </pre> <p>Ответ:</p> <pre> {   "StatusCode": 0,   "Message": "OK",   "Data": {     "QrPaymentId": 15   } } </pre>			

**4.6.3. Отмена счета на оплату**

Описание	Отмена счета на оплату			
Поставщик	Kaspi Pay			
Потребитель	Онлайн-касса			
Метод	POST			
Адрес	https://mtokentest.kaspi.kz:8545/r3/v01/remote/cancel			
Входные параметры	<b>Наименование</b>	<b>Тип</b>	<b>Обязательный</b>	<b>Описание</b>
	OrganizationBin	String	Да	БИН/ИИН организации

**Третья схема** Вариант с усиленной безопасностью

	QrPaymentId	Int64	Да	Идентификатор покупки
	DeviceToken	Int64	Да	Идентификатор добавленного в Kaspi Pay устройства
Выходные параметры	Status: String – статус покупки			
Возможные StatusCode	0, -999, -1401, -1404, -1501, -99000001, -99000002, 990000018			
Пример	<pre>https://mtokentest.kaspi.kz:8545/r3/v01/remote/cancel Запрос: {   "OrganizationBin": "180340021791",   "QrPaymentId": 15,   "DeviceToken": 321 }  Ответ: {   "StatusCode": 0,   "Message": "OK",   "Data": {     "Status": "RemotePaymentCanceled"   } }</pre>			

## Служебные методы

### 5. Служебные методы

#### 5.1. Проверка доступности (пинг)

Описание	Проверка доступности (работоспособности) сервиса
Поставщик	Kaspi Pay
Потребитель	Партнер
Метод	GET
Адреса по уровню доступа	Облегченный https://mtokentest.kaspi.kz:8543/r1/v01/health/ping
	Стандартный https://mtokentest.kaspi.kz:8544/r2/v01/health/ping
	Усиленный https://mtokentest.kaspi.kz:8545/r3/v01/health/ping
Входные параметры	Нет
Выходные параметры	Нет
Возможные StatusCode	0, -999, -10000
Пример	https://mtokentest.kaspi.kz:8543/r1/v01/health/ping Ответ: { "StatusCode": 0, "Message": "OK" }

#### 5.2. Эмуляция сканирования QR

Описание	Метод, эмулирующий сканирование QR
Поставщик	Kaspi Pay
Потребитель	Партнер
Метод	POST
Адреса по уровню доступа	Облегченный https://mtokentest.kaspi.kz:8543/r1/v01/test/payment/scan
	Стандартный https://mtokentest.kaspi.kz:8544/r2/v01/test/payment/scan
	Усиленный https://mtokentest.kaspi.kz:8545/r3/v01/test/payment/scan
Входные параметры	{ "qrPaymentId": "string" }
Выходные параметры	{ "StatusCode": 0, "message": "string" }
Возможные StatusCode	0, -999, -10000

#### 5.3. Эмуляция подтверждения покупки клиентом

Описание	Метод, эмулирующий подтверждение покупки клиентом
Поставщик	Kaspi Pay
Потребитель	Партнер
Метод	POST
Адреса по уровню доступа	Облегченный https://mtokentest.kaspi.kz:8543/r1/v01/test/payment/confirm

## Служебные методы

	Стандартный <a href="https://mtokentest.kaspi.kz:8544/r2/v01/test/payment/confirm">https://mtokentest.kaspi.kz:8544/r2/v01/test/payment/confirm</a>
	Усиленный <a href="https://mtokentest.kaspi.kz:8545/r3/v01/test/payment/confirm">https://mtokentest.kaspi.kz:8545/r3/v01/test/payment/confirm</a>
Входные параметры	{ "qrPaymentId": "string" }
Выходные параметры	{ "StatusCode": 0, "message": "string" }
Возможные StatusCode	0, -999, -10000

### 5.4. Эмуляция ошибки при сканировании QR

Описание	Метод, который переводит статус операции в Error при эмуляции сканирования QR.
Поставщик	Kaspi Pay
Потребитель	Партнер
Метод	POST
Адреса по уровню доступа	Облегченный <a href="https://mtokentest.kaspi.kz:8543/r1/v01/test/payment/scanerror">https://mtokentest.kaspi.kz:8543/r1/v01/test/payment/scanerror</a>
	Стандартный <a href="https://mtokentest.kaspi.kz:8544/r2/v01/test/payment/scanerror">https://mtokentest.kaspi.kz:8544/r2/v01/test/payment/scanerror</a>
	Усиленный <a href="https://mtokentest.kaspi.kz:8545/r3/v01/test/payment/scanerror">https://mtokentest.kaspi.kz:8545/r3/v01/test/payment/scanerror</a>
Входные параметры	{ "qrPaymentId": "string" }
Выходные параметры	{ "StatusCode": 0, "message": "string" }
Возможные StatusCode	0, -999, -10000

### 5.5. Эмуляция ошибки при подтверждении покупки клиентом

Описание	Метод, который переводит статус операции (2.2.2) в Error при подтверждении покупки клиентом. <b>Важно:</b> Данный метод вызывается после успешной эмуляции сканирования QR (5.3).
Поставщик	Kaspi Pay
Потребитель	Партнер
Метод	POST
Адреса по уровню доступа	Облегченный <a href="https://mtokentest.kaspi.kz:8543/r1/v01/test/payment/confirmerror">https://mtokentest.kaspi.kz:8543/r1/v01/test/payment/confirmerror</a>
	Стандартный <a href="https://mtokentest.kaspi.kz:8544/r2/v01/test/payment/confirmerror">https://mtokentest.kaspi.kz:8544/r2/v01/test/payment/confirmerror</a>
	Усиленный <a href="https://mtokentest.kaspi.kz:8545/r3/v01/test/payment/confirmerror">https://mtokentest.kaspi.kz:8545/r3/v01/test/payment/confirmerror</a>
Входные параметры	{ "qrPaymentId": "string" }



**Служебные методы**


Выходные параметры	{ "StatusCode": 0, "message": "string" }
Возможные StatusCode	0, -999, -10000

### Приложение 1. Импорт (установка) и использование сертификатов

**kbs\_root\_ca.crt** – корневой сертификат Центра Сертификации Kaspi Pay, его нужно импортировать в хранилище «Доверенные корневые центры сертификации (Trusted Root Certification Authorities)».

**kbs\_ca.crt** – сертификат промежуточного Центра Сертификации Kaspi Pay, его нужно импортировать в хранилище «Промежуточные центры сертификации (Intermediate Certification Authorities)».

Импорт сертификатов, в ОС Windows, осуществляется путем двойного клика на файл сертификата и нажатия на кнопку «Установить сертификат...». Далее, указать расположение «Локальный компьютер» и выбрать соответствующее хранилище сертификатов.


 **Важно!** Сертификаты должны быть установлены на сервере с которого будут отправляться запросы в API. Не выбирать опцию «Автоматически выбрать хранилище на основе типа сертификата», а выбрать хранилище вручную!

#### ДЛЯ ВЫХОДА В ПРОДАКШН

1. Создать закрытый ключ и файл запроса на выпуск сертификата (Certificate Signing Request – файл \*.csr). О том, как создать закрытый ключ и файл запроса с помощью утилиты openssl написано в Приложении 2.

2. Отправить файл запроса на выпуск сертификата в банк (файл \*.csr) на адрес электронной почты **cert.request@kaspi.kz** (с адреса электронной почты, указанного в договоре)

3. В ответном письме от банка придет архив с выпущенным клиентским сертификатом, корневым (kbs\_root\_ca.crt) и промежуточным (kbs\_ca.crt) сертификатами центра сертификации банка. Корневой и промежуточный сертификаты нужно установить (импортировать) на машину, с которой будут отправляться запросы в API. Также в письме будет **адрес к продуктовому API**.

 **Важно!** Сертификат привязывается к организации по БИН и выпускается сроком на **5 лет!** По истечению срока потребуется перевыпуск сертификата с последующей заменой старого на новый. Поэтому **при реализации клиентского приложения необходимо** предусмотреть возможность замены и дальнейшее использование обновленного сертификата без дополнительных доработок и привлечения разработчиков. Это же касается и **пароля приватного ключа**, так как он используется при запросах в API.

4. Реализовать в клиентском приложении использование клиентского сертификата при отправке запросов в API. (реализация индивидуально зависит от используемых средств и языка разработки)

5. Опционально: реализовать в клиентском мобильном приложении механизм пиннинга корневого сертификата.

6. Проверить подключение после разработки и настройки клиентского приложения.

## Приложение 2

### Приложение 2. Создание запроса на выпуск сертификата с использованием утилиты openssl

Для того чтобы создать закрытый ключ (**private.key**) и сформировать файл запроса на выпуск сертификата (Certificate Signing Request – файл \*.csr), необходимо выполнить команду утилиты openssl. В запросе на подпись клиентского сертификата будет находиться только открытый ключ партнера.

ВАЖНО! Перед выполнением команды утилиты openssl, указанной ниже, необходимо откорректировать текст команды и указать корректные пути к самой утилите и файлам, указанным в команде.

```
openssl req -newkey rsa:2048 -keyout private.key -out certrequest.csr -addext  
'keyUsage = digitalSignature, nonRepudiation, keyEncipherment,  
dataEncipherment' -addext 'extendedKeyUsage = clientAuth'
```

После запуска команды необходимо будет указать значения, которые будут записаны в клиентском сертификате. Диалог выглядит примерно так (цифры указаны для удобства и означают номера строк):

```
1) Generating a RSA private key
2) .....+++++
3) .....+++++
4) writing new private key to 'private.key'
5) Enter PEM pass phrase:
6) Verifying - Enter PEM pass phrase:
7) -----
8) You are about to be asked to enter information that will be incorporated
9) into your certificate request.
10) What you are about to enter is what is called a Distinguished Name or a
    DN.
11) There are quite a few fields but you can leave some blank
12) For some fields there will be a default value,
13) If you enter '.', the field will be left blank.
14) -----
15) Country Name (2 letter code) [XX]:KZ
16) State or Province Name (full name) []:Almaty Province
17) Locality Name (eg, city) []:Almaty
18) Organization Name (eg, company) [Default Company Ltd]:Contoso Ltd.
19) Organizational Unit Name (eg, section) []:
20) Common Name (eg, your name or your server's hostname) []:
21) Email Address []:
22)
23) Please enter the following 'extra' attributes
24) to be sent with your certificate request
25) A challenge password []:
26) An optional company name []:
```

В строке 5 нужно придумать и указать пароль доступа к закрытому ключу. Этот ключ понадобится при выполнении команды из раздела «Создание контейнера с закрытым ключом и клиентским сертификатом»

В строке 6 нужно повторить ввод пароля доступа к закрытому ключу.

## Приложение 2

В строке 15 нужно указать название страны. KZ

В строке 16 нужно указать название региона, где расположена организация партнера, например: Almaty Province;

В строке 17 нужно указать название населенного пункта, например: Almaty;

В строке 18 нужно указать юридическое название организации партнера на английском языке. Если у организации партнера нет названия на английском языке, то при формировании запроса необходимо использовать латинскую транслитерацию. Например: Contoso Ltd;

В строке 20 нужно указать адрес хоста организации, к которому будет производиться подключение, например (IP адрес или DNS имя точки подключения);

В строке 21 нужно указать адрес электронной почты контактного лица организации партнера;

В строках 19,25,26 просто нажать «Enter», оставив значения незаполненными.

В результате выполнения этой команды создадутся файлы **private.key** и **certrequest.csr**. В файле **private.key** будет находиться закрытый ключ, а файл **certrequest.csr** будет содержать запрос на подпись клиентского сертификата.

### Создание PFX контейнера с закрытым ключом и клиентским сертификатом с использованием утилиты openssl

После получения архива с сертификатами, необходимо создать контейнер с закрытым ключом и клиентским сертификатом из архива. Для этого нужно выполнить команду утилиты openssl и создать файл **certfile.pfx** (контейнер, в котором будут содержаться его закрытый ключ и его клиентский сертификат **certfile.cer**)

**Важно!** Перед выполнением команды утилиты openssl, указанной ниже, необходимо откорректировать текст команды и указать корректные пути к самой утилите и файлам, указанным в команде.

```
openssl pkcs12 -export -in certfile.cer -inkey private.key -out certfile.pfx
```

Для установки TLS-подключения и вызова методов Public Qr API, в клиентском приложении нужно использовать файл **certfile.pfx**

Приложение 3. Значения StatusCode

StatusCode	Описание
0	Успешный статус операции
-10000	Отсутствует сертификат клиента
-1501	Устройство с заданным идентификатором не найдено
-1502	Устройство не активно (отключено или удалено)
-1503	Устройство уже добавлено в другую торговую точку
- 1601	Покупка не найдена
-14000002	Отсутствуют торговые точки, необходимо создать торговую точку в приложении Kaspi Pay
-99000002	Торговая точка не найдена
-99000005	Сумма возврата не может превышать сумму покупки
-99000006	Ошибка возврата, необходимо попробовать еще раз и при повторении ошибки обратиться в банк
990000018	Торговая точка отключена
990000026	Торговая точка не принимает оплату с QR
990000028	Указана неверная сумма операции
990000033	Нет доступных методов оплаты
-990000001	Покупка с заданным идентификатором не найдена
-990000002	Торговая точка не найдена
-990000003	Торговая точка покупки не соответствует текущему устройству
-990000011	Невозможно вернуть покупку (несоответствующий статус покупки)
-990000020	Частичный возврат невозможен
-999	Сервис временно недоступен

Приложение 4. Статусы покупки

Статус	Значение	Конечный
QrTokenCreated	Создан новый токен	Нет
Wait	Клиент отсканировал QR, ожидается подтверждение	Нет
Processed	Покупка подтверждена клиентом и успешно проведена	Да
Error	Покупка отменена клиентом, или возникла ошибка во время проведения операции	Да

Приложение 5. Статусы возврата

Статус	Значение	Конечный
QrTokenCreated	Создан новый токен	Нет
CustomerIdentityObtained	Клиент отсканировал QR	Да
Error	Ошибка во время проведения возврата	Да