

Интерфейс REST

Введение

REST - простой способ передачи данных по сети. Данные передаются с помощью HTTP-запросов (GET или POST), содержащих набор параметров. Результат обработки запроса возвращается в виде объекта JSON (см. пример результата ниже).

```
{ "errorCode": "12", "errorMessage": "Empty amount" }
```

Аутентификация продавца

Для аутентификации продавца при обращении к платёжному шлюзу можно использовать один из следующих способов.

- С помощью логина и пароля служебной учётной записи (учётная запись с суффиксом - **api**), полученной при регистрации. Значения имени и пароля передаются в параметрах `userName` и `password` соответственно (см. таблицу ниже).

Название	Тип	Обязательно	Описание
userName	AN..30	См. описание	Логин служебной учётной записи продавца. При передаче логина и пароля для аутентификации в платёжном шлюзе параметр <code>token</code> передавать не нужно.
password	AN..200	См. описание	Пароль служебной учётной записи продавца. При передаче логина и пароля для аутентификации в платёжном шлюзе параметр <code>token</code> передавать не нужно.

- С помощью специального токена - его значение можно получить у технической поддержки. Значение токена передаётся в параметре `token` (см. таблицу ниже).

Авторизация посредством передачи `token` возможна лишь в случае с нефинансовыми запросами:

- Регистрация заказа ([registerOrder.do](#))
- Регистрация заказа с предавторизацией ([registerPreAuth.do](#))
- Получение статуса заказа ([getOrderStatusExtended.do](#))

Во всех остальных запросах необходимо передавать обязательные параметры `userName` и `password`.

token	AN..256	См. описание	Значение, которое используется для аутентификации продавца при отправке запросов в платёжный шлюз. При передаче этого параметра параметры <code>userName</code> и <code>password</code> передавать не нужно. Чтобы получить открытый ключ, обратитесь в техническую поддержку.
-------	---------	--------------	---

Пользуясь протоколом REST, следует отправлять запросы с типом `application/x-www-form-urlencoded`, а не `multipart/form-data`.

Передача авторизационных данных (логин пользователя / пароль) можно передавать как в теле запроса, так и в заголовке.

- Пример передачи данных для авторизации в заголовке запроса:

<https://3dsec.sberbank.ru/payment/rest/getOrderStatusExtended.do?userName=username-api&password=password&orderId=961e34e7-e897-7d22-8b62-19650008f9da>

- Пример передачи данных для авторизации в теле запроса:

```
{
  "userName": "username-api",
  "password": "password",
  "orderId": "961e34e7-e897-7d22-8b62-19650008f9da"
}
```

Запросы

Запросы должны соответствовать следующим требованиям:

- все текстовые поля должны иметь кодировку Юникод (UTF-8);
- специальные символы в запросе REST должны быть экранированы в соответствии с URL-кодом, например, пароль `qwe?rt%y` должен передаваться в виде `qwe%3Frt%25y`. Дополнительную информацию об экранировании в запросах и таблицу символов см. по [адресу](#).

Никакие передаваемые поля API не могут содержать части `html`, `javascript` и другого кода. Поля могут содержать исключительно текст и ссылки.

При несоблюдении данного требования система будет распознавать такие случаи как нарушение, а такие запросы будут отклоняться.

Обратная совместимость API не нарушается, если:

- появляется новый необязательный параметр в запросе;
- появляется новый параметр в ответе;
- добавляются новые значения в `enum`;
- меняется порядок параметров в ответе.

URL-адреса для доступа к запросам REST представлены в таблице ниже.

Название запроса	URL
Регистрация заказа	https://3dsec.sberbank.ru/payment/rest/register.do
Регистрация заказа с предавторизацией	https://3dsec.sberbank.ru/payment/rest/registerPreAuth.do
Запрос завершения оплаты заказа	https://3dsec.sberbank.ru/payment/rest/deposit.do
Запрос отмены оплаты заказа	https://3dsec.sberbank.ru/payment/rest/reverse.do
Запрос возврата средств оплаты заказа	https://3dsec.sberbank.ru/payment/rest/refund.do
Запрос возврата средств по устаревшему заказу сроком более двух лет	https://3dsec.sberbank.ru/payment/rest/instantRefund.do
Возврат средств без привязки к товарной позиции	https://3dsec.sberbank.ru/payment/rest/processRawSumRefund.do
Возврат на произвольную сумму по одной товарной позиции	https://3dsec.sberbank.ru/payment/rest/processRawPositionRefund.do
Получение статуса заказа	https://3dsec.sberbank.ru/payment/rest/getOrderStatusExtended.do
Запрос проверки вовлечённости карты в 3DS	https://3dsec.sberbank.ru/payment/rest/verifyEnrollment.do
Запрос оплаты через Apple Pay	https://3dsec.sberbank.ru/payment/applepay/payment.do
Запрос рекуррентных платежей Apple Pay	https://3dsec.sberbank.ru/payment/recurrentPayment.do
Запрос оплаты через Samsung Pay	https://3dsec.sberbank.ru/payment/samsung/payment.do
Запрос оплаты через Samsung Pay (web)	https://3dsec.sberbank.ru/payment/samsungWeb/payment.do
Запрос оплаты через Google Pay	https://3dsec.sberbank.ru/payment/google/payment.do
Запрос оплаты через Mir Pay	https://3dsec.sberbank.ru/payment/mir/payment.do
Запрос оплаты через Mir Pay с расшифровкой платежных данных на стороне Мерчанта	https://3dsec.sberbank.ru/payment/mir/paymentDirect.do
Запрос отмены неоплаченного заказа	https://3dsec.sberbank.ru/payment/rest/decline.do
Запрос сведений о кассовом чеке (Для продавцов, подключённых к оператору фискальных данных.)	https://3dsec.sberbank.ru/payment/rest/getReceiptStatus.do
Запрос деактивации связи	https://3dsec.sberbank.ru/payment/rest/unBindCard.do
Запрос активации связи	https://3dsec.sberbank.ru/payment/rest/bindCard.do
Запрос списка всех связей клиента	https://3dsec.sberbank.ru/payment/rest/getBindings.do
Запрос списка связей определённой банковской карты	https://3dsec.sberbank.ru/payment/rest/getBindingsByCardOrId.do

Название запроса	URL
Запрос изменения срока действия связи	https://3dsec.sberbank.ru/payment/rest/extendBinding.do
Запрос списка связей клиента с дубликатами	https://3dsec.sberbank.ru/payment/rest/getAllBindings.do
Запрос создания связи без проведения платежа	https://3dsec.sberbank.ru/payment/rest/createBindingNoPayment.do
Запрос передачи данных чека	https://3dsec.sberbank.ru/fes-nsfk-proxy/externalReceipt

Особенности тестирования запросов REST в браузере

Если тестирование REST-запросов происходит в том же браузере, в котором выполнен вход в личный кабинет, то выполнение любого REST-запроса приведёт к тому, что в личном кабинете будет отображаться ошибка. Чтобы восстановить работу личного кабинета:

- выполните выход из личного кабинета и снова зайдите в него;
- если предыдущий способ не работает, удалите cookie-файлы и снова войдите в личный кабинет.

Подобной ситуации можно избежать одним из следующих способов:

- используйте для REST-запросов режим «инкогнито», а работу в личном кабинете проводите в окне браузера в обычном режиме;
- для тестирования REST-запросов и работы в личном кабинете используйте разные браузеры.

Возможные сценарии взаимодействия

В соответствии с российским законодательством продавцы должны отправлять электронные версии чеков оператору фискальных данных через интернет. Существует возможность упростить эту процедуру: вы будете отправлять необходимые данные в запросах, а регистрацию чеков на контрольно-кассовой технике будет осуществлять «Сбербанк».

- [Сценарии взаимодействия при обычной оплате](#) (регистрация чеков реализуется продавцом самостоятельно)
- [Сценарии взаимодействия при регистрации чеков на контрольно-кассовой технике](#) (регистрация чеков производится «Сбербанком»)
 - [Стандартный API](#)
 - [Apple Pay](#)
 - [Google Pay](#)
 - [Samsung Pay](#)
 - [Mir Pay](#)
 - [Mir Pay \(Расшифровка платежных данных на стороне Мерчанта\)](#)
- [Оплата с предоставлением возможности интернет-кредитования](#) (сценарии взаимодействия при оплате с предоставлением покупателю возможности получить кредит)
- [Сценарии взаимодействия при оплате по связкам](#)

Обычная оплата

Оплата с указанием карточных данных на платёжной странице

	Одностадийная оплата	Двухстадийная оплата
1	Плательщик выбирает товар или услугу на ресурсе продавца и выбирает способ оплаты банковской картой.	
2	Продавец отправляет в платёжный шлюз запрос на регистрацию заказа: • register.do. В том числе в запросе передаются следующие параметры: • <code>returnUrl</code> - URL-адрес, на который будет перенаправлен плательщик в случае успешного платежа; • <code>failUrl</code> - URL-адрес, на который будет перенаправлен плательщик в случае неуспешного платежа.	Продавец отправляет в платёжный шлюз запрос на регистрацию заказа с предавторизацией: • registerPreAuth.do. В том числе в запросе передаются следующие параметры: • <code>amount</code> - сумма списания; • <code>orderId</code> - номер заказа в системе магазина; • <code>returnUrl</code> - URL-адрес, на который будет перенаправлен плательщик в случае успешного платежа; • <code>failUrl</code> - URL-адрес, на который будет перенаправлен плательщик в случае неуспешного платежа.
3	Платёжный шлюз в ответе в числе прочего возвращает параметры: • <code>orderId</code> - уникальный идентификатор заказа в платёжной системе; • <code>formUrl</code> - URL-адрес формы ввода данных банковской карты (платёжная форма).	
4	Продавец перенаправляет плательщику URL-адрес, указанный в параметре <code>formUrl</code> ответа.	
5	В браузере плательщика отображается форма ввода данных банковской карты. Плательщик заполняет форму и отправляет данные на сервер платёжного шлюза.	
6	Дальнейшие действия зависят от того, поддерживает ли карта покупателя 3-D Secure: • поддержка 3-D Secure отсутствует - переход к следующему шагу процедуры; • есть поддержка 3-D Secure - платёжный шлюз перенаправляет плательщика на форму аутентификации (чаще всего это проверка с помощью SMS-сообщения), расположенную на сервере контроля доступа, принадлежащего банку-эмитенту - в случае успешной аутентификации выполняется следующий шаг сценария.	
7	Платёжный шлюз списывает деньги со счёта плательщика.	Платёжный шлюз производит холдирование (удержание) средств на счёте плательщика.
8	После проведения оплаты платёжный шлюз перенаправляет клиента на URL-адрес возврата (указанный магазином в запросе на регистрацию).	
9	Браузер клиента запрашивает страницу с результатами оплаты у магазина.	
10	Продавец запрашивает у платёжного шлюза статус оплаты заказа, используя уникальный идентификатор, полученный при регистрации заказа в параметре <code>orderId</code>: • Расширенный запрос состояния заказа (getStatusExtended.do).	
11	Платёжный шлюз возвращает статус оплаты и продавец передаёт в браузер клиента страницу с результатами оплаты.	

	Одностадийная оплата	Двухстадийная оплата
12	Не актуально	Для списания средств со счёта клиента продавец должен направить в платёжный шлюз запрос завершения оплаты: • deposit.do.
13	Не актуально	Платёжный шлюз возвращает результат обработки запроса. Статус заказа не возвращается. Для получения статуса заказа необходимо отправить в шлюз запрос состояния заказа: • Расширенный запрос состояния заказа (getOrderStatusExtended.do).

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса заказа.

Отмена платежа

Отмена оплаты заказа доступна при наличии соответствующих прав по согласованию со «Сбербанком».

- При одностадийных платежах отмена платежа возможна для заказов в состоянии **Завершён/Deposited**, но только до закрытия дня, т. е. до 24:00 (по московскому времени) того дня, когда была произведена оплата.
- При двухстадийных платежах отмену платежа можно выполнить для заказа в состоянии **Подтверждён/Approved**.

В таблице ниже представлена схема взаимодействий продавца и платёжного шлюза при отмене оплаты.

1	Продавец делает запрос на отмену платежа: • reverse.do.
2	После получения успешного ответа платёжного шлюза продавец делает запрос статуса заказа: • Расширенный запрос состояния заказа (getOrderStatusExtended.do);

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса заказа.

Возврат средств плательщику

Платёжный шлюз позволяет осуществлять полный или частичный возврат средств по оплаченным заказам.

Возврат средств возможен для платежей в состоянии **Завершён/Deposited**.

В таблице ниже представлена схема взаимодействий продавца и платёжного шлюза при возврате средств покупателю.

1	Продавец делает запрос на возврат средств покупателю: • refund.do.
2	После получения успешного ответа платёжного шлюза продавец делает запрос статуса заказа: • Расширенный запрос состояния заказа (getOrderStatusExtended.do).

При наличии соответствующей пермиссии также доступен [Запрос возврата средств по устаревшему заказу сроком более двух лет](#).

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса заказа.

Оплата с регистрацией чеков на контрольно-кассовой технике

Стандартный API

Оплата с передачей корзины и данных для регистрации чека

	Одностадийная оплата	Двухстадийная оплата
1	Платательщик выбирает товар или услугу на ресурсе продавца и выбирает способ оплаты банковской картой.	

	Одностадийная оплата	Двухстадийная оплата
2	Продавец отправляет в платёжный шлюз запрос на регистрацию заказа: • register.do. В том числе в запросе передаются следующие параметры: • returnUrl - URL-адрес, на который будет перенаправлен плательщик в случае успешного платежа; • failUrl - URL-адрес, на который будет перенаправлен плательщик в случае неуспешного платежа. • orderBundle - блок, содержащий корзину товаров и данные для регистрации чека; • taxSystem - идентификатор системы налогообложения.	Продавец отправляет в платёжный шлюз запрос на регистрацию заказа с предавторизацией: • registerPreAuth.do. В том числе в запросе передаются следующие параметры: • amount - сумма списания; • orderNumber - номер заказа в системе магазина; • returnUrl - URL-адрес, на который будет перенаправлен плательщик в случае успешного платежа; • failUrl - URL-адрес, на который будет перенаправлен плательщик в случае неуспешного платежа. • orderBundle - блок, содержащий корзину товаров и данные для регистрации чека; • taxSystem - идентификатор системы налогообложения.
3	Платёжный шлюз в ответе в числе прочего возвращает параметры: • orderId - уникальный идентификатор заказа в платёжной системе; • formUrl - URL-адрес формы ввода данных банковской карты (платёжная форма).	
4	Продавец перенаправляет плательщику URL-адрес, указанный в параметре formUrl ответа.	
5	В браузере плательщика отображается форма ввода данных банковской карты. Плательщик заполняет форму и отправляет данные на сервер платёжного шлюза.	
6	Дальнейшие действия зависят от того, поддерживает ли карта покупателя 3-D Secure: • поддержка 3-D Secure отсутствует - переход к следующему шагу процедуры; • есть поддержка 3-D Secure - платёжный шлюз перенаправляет плательщика на форму аутентификации (чаще всего это проверка с помощью SMS-сообщения), расположенную на сервере контроля доступа, принадлежащего банку-эмитенту - в случае успешной аутентификации выполняется следующий шаг сценария.	
7	Платёжный шлюз списывает деньги со счёта плательщика.	Платёжный шлюз производит холдирование (удержание) средств на счёте плательщика.
8	После проведения оплаты платёжный шлюз перенаправляет клиента на URL-адрес возврата.	
9	Браузер клиента запрашивает страницу с результатами оплаты у магазина.	
10	Продавец запрашивает у платёжного шлюза статус оплаты заказа, используя уникальный идентификатор, полученный при регистрации заказа в параметре orderId: • getOrderStatusExtended.do.	
11	Платёжный шлюз возвращает статус оплаты и продавец передаёт в браузер клиента страницу с результатами оплаты.	

	Одностадийная оплата	Двухстадийная оплата
12	Не актуально	Для списания средств со счёта клиента продавец должен направить в платёжный шлюз запрос завершения оплаты: - завершение с оплатой полной суммой: deposit.do; - завершение с оплатой части суммы: deposit.do.
13	Не актуально	Платёжный шлюз возвращает результат обработки запроса. Статус заказа не возвращается. Для получения статуса заказа необходимо отправить в шлюз запрос состояния заказа: getOrderStatusExtended.do.
14	Если клиент получил товар или услугу не в момент оплаты, необходимо отправить запрос на создание чека закрытия .	
15	Если при использовании функционала ЭС используется фискализация на стороне мерчанта, необходимо отправить запрос передачи данных чека externalReceipt.do .	

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса заказа.

Чтобы получить сведения о кассовом чеке (например, чтобы передать эти данные покупателю) используйте запрос [getReceiptStatus.do](#). В настоящее время этот запрос можно выполнить только через интерфейс REST.

Отмена платежа

Отмена оплаты заказа доступна при наличии соответствующих прав по согласованию со «Сбербанком».

- При одностадийных платежах отмена платежа возможна для заказов в состоянии **Завершён/Deposited**, но только до закрытия дня, т. е. до 24:00 (по московскому времени) того дня, когда была произведена оплата.
- При двухстадийных платежах отмену платежа можно выполнить для заказа в состоянии **Подтверждён/Approved**.

В таблице ниже представлена схема взаимодействий продавца и платёжного шлюза при отмене оплаты.

1	Продавец делает запрос на отмену платежа: reverse.do.
---	--

2	После получения успешного ответа платёжного шлюза продавец делает запрос статуса заказа: • Расширенный запрос состояния заказа (getOrderStatusExtended.do);
---	---

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса заказа.

Возврат средств плательщику

Платёжный шлюз позволяет осуществлять полный или частичный возврат средств по оплаченным заказам.

Возврат средств возможен для платежей в состоянии **Завершён/Deposited**.

В таблице ниже представлена схема взаимодействий продавца и платёжного шлюза при возврате средств покупателю.

1	Продавец делает запрос на возврат средств покупателю: • возврат полной суммы: ◦ refund.do; • частичный возврат: ◦ refund.do.
2	После получения успешного ответа платёжного шлюза продавец делает запрос статуса заказа: • getOrderStatusExtended.do.
3	Если для заказа был осуществлен частичный возврат , то продавец делает запрос закрытия чека: • closeOfdReceipt.do.

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса заказа.

Apple Pay

При оплате с использованием Apple Pay взаимодействие происходит по следующей схеме.

Одностадийный платёж	Двухстадийный платёж
----------------------	----------------------

	Одностадийный платёж	Двухстадийный платёж
1	Пользователь в мобильном приложении выбирает вариант оплаты с помощью Apple Pay.	
2	Сведения о платеже направляются на обработку в систему Apple Pay.	
3	Для обработки данных о платеже в системе Apple Pay создаётся объект <code>RKPaymentToken Object</code> , который содержит свойство <code>paymentData</code> (здесь и далее см. документацию Apple Pay).	
4	Apple Pay направляет продавцу (мобильному приложению) ответ.	
5	Продавец извлекает из полученного объекта <code>RKPaymentToken Object</code> свойство <code>paymentData</code> и кодирует его содержимое в Base64.	
6	Продавец создаёт запрос на оплату, содержащий в том числе свойство <code>paymentData</code> , полученное из ответа системы Apple Pay и закодированное в Base64, и отправляет его на обработку в платёжный шлюз. Для оплаты через Apple Pay используется следующий запрос: payment.do (интерфейс REST). При этом параметр <code>preAuth</code> либо не передаётся, либо имеет значение <code>false</code> (ложь).	Продавец создаёт запрос на оплату, содержащий в том числе свойство <code>paymentData</code> , полученное из ответа системы Apple Pay и закодированное в Base64, и отправляет его на обработку в платёжный шлюз. Для оплаты через Apple Pay используется следующий запрос: payment.do (интерфейс REST). При этом в запросе передаётся параметр <code>preAuth</code> , который имеет значение <code>true</code> (истина).
7	Платёжный шлюз обрабатывает запрос и возвращает ответ с результатом: интерфейс REST.	
8	Мобильное приложение отображает пользователю результат оплаты.	
9	Продавец запрашивает у платёжного шлюза статус оплаты заказа, используя уникальный идентификатор, полученный при регистрации заказа в параметре <code>orderId</code> : getOrderStatusExtended.do (интерфейс REST).	
10	Не актуально	Для списания средств со счёта клиента продавец должен направить в платёжный шлюз запрос завершения оплаты: deposit.do (интерфейс REST).
11	Не актуально	Платёжный шлюз возвращает результат обработки запроса. Статус заказа не возвращается. Для получения статуса заказа необходимо отправить в шлюз запрос состояния заказа: getOrderStatusExtended.do (интерфейс REST).
12	Если клиент получил товар или услугу не в момент оплаты, необходимо отправить запрос на создание чека закрытия .	

Google Pay

- Сведения для разработчиков представлены на [сайте Google](#).
- Приложение должно быть оформлено в соответствии с [руководством по использованию бренда](#).

При отправке запроса в Google Pay используйте следующие параметры:

- gateway: sberbank;
- gatewayMerchantId: <наименование вашей организации в системе платёжного шлюза>.

При оплате с использованием Google Pay через мобильное приложение взаимодействие происходит по следующей схеме.

	Одностадийный платёж	Двухстадийный платёж
1	Пользователь выбирает способ оплаты Google Pay.	
2	Приложение запрашивает Google Pay маскированные данные платёжной карты.	
3	Google Pay возвращает в приложение маскированные данные платёжной карты.	
4	Приложение отображает клиенту маскированные данные карты, добавленной в Google Pay.	
5	Пользователь подтверждает оплату.	
6	Приложение запрашивает Google Pay зашифрованные данные платёжной карты.	
7	Google шифрует данные, используя открытый ключ - соответствующий ему закрытый ключ расположен в платёжном шлюзе.	
8	Google Pay возвращает в приложение зашифрованные данные о платеже.	
9	Приложение отправляет в платёжный шлюз запрос на оплату, указывая полученный от Google Pay токен: • интерфейс REST (payment.do). При этом параметр preAuth либо не передаётся, либо имеет значение false (ложь).	Приложение отправляет в платёжный шлюз запрос на оплату, указывая полученный от Google Pay токен: • интерфейс REST (payment.do). При этом в запросе передаётся параметр preAuth, который имеет значение true (истина).
10	Платёжный шлюз расшифровывает полученный токен и производит оплату.	
11	Платёжный шлюз возвращает результат оплаты в приложение.	
12	Приложение отображает результат покупки плательщику.	
13	Продавец запрашивает у платёжного шлюза статус оплаты заказа, используя уникальный идентификатор, полученный при регистрации заказа в параметре orderId: • интерфейс REST (getOrderStatusExtended.do).	
14	Не актуально	Для списания средств со счёта клиента продавец должен направить в платёжный шлюз запрос завершения оплаты: • интерфейс REST (deposit.do).
15	Не актуально	Платёжный шлюз возвращает результат обработки запроса. Статус заказа не возвращается. Для получения статуса заказа необходимо отправить в шлюз запрос состояния заказа: • интерфейс REST (getOrderStatusExtended.do).
16	Если клиент получил товар или услугу не в момент оплаты, необходимо отправить запрос на создание чека закрытия .	

При оплате с использованием Samsung Pay взаимодействие происходит по следующей схеме.

	Одностадийный платёж	Двухстадийный платёж
1	Пользователь выбирает способ оплаты Samsung Pay.	
2	Приложение отправляет сведения о платеже в Samsung.	
3	Samsung отправляет в приложение ответ содержащий, среди прочего, параметр 3ds.data с зашифрованными данными о платеже.	
4	Продавец отправляет в платёжный шлюз запрос на оплату: • интерфейс REST (payment.do). При этом: • параметр paymentToken включает содержимое 3ds.data, полученное от Samsung; • параметр preAuth либо не передаётся, либо имеет значение false (ложь).	Продавец отправляет в платёжный шлюз запрос на оплату: • интерфейс REST (payment.do). При этом: • параметр paymentToken включает содержимое 3ds.data, полученное от Samsung; • в запросе передаётся параметр preAuth, который имеет значение true (истина).
5	Платёжный шлюз расшифровывает полученный токен и производит оплату.	
6	Платёжный шлюз возвращает в приложение результат оплаты, который отображается на экране мобильного устройства.	
7	Продавец запрашивает у платёжного шлюза статус оплаты заказа, используя уникальный идентификатор, полученный при регистрации заказа в параметре orderId: • интерфейс REST (getOrderStatusExtended.do).	
8	Не актуально	Для списания средств со счёта клиента продавец должен направить в платёжный шлюз запрос завершения оплаты: • интерфейс REST (deposit.do).
9	Не актуально	Платёжный шлюз возвращает результат обработки запроса. Статус заказа не возвращается. Для получения статуса заказа необходимо отправить в шлюз запрос состояния заказа: • интерфейс REST (getOrderStatusExtended.do).
10	Если клиент получил товар или услугу не в момент оплаты, необходимо отправить запрос на создание чека закрытия .	

Mir Pay

При оплате с использованием Mir Pay взаимодействие происходит по следующей схеме.

	Одностадийный платёж	Двухстадийный платёж
1	Пользователь выбирает способ оплаты Mir Pay.	
2	Мерчант через MirPaySDK запускает сценарий подготовки In-Application операции в приложении Mir Pay/Мерчант через Deeplink или Universal Link запускает сценарий подготовки In-Application операции в приложении Mir Pay.	
3	Mir Pay отображает список карт клиенту.	
4	Клиент выбирает карту для оплаты в приложении MirPay.	

	Одностадийный платёж	Двухстадийный платёж
5	Mir Pay формирует данные платежа и отправляет зашифрованные данные платежа мерчанту.	
6	Мерчант отправляет в платежный шлюз запрос на оплату: • /mir/payment.do. При этом: • параметр preAuth либо не передаётся, либо имеет значение false (ложь).	Мерчант отправляет в платежный шлюз запрос на оплату: • /mir/payment.do; При этом: • в запросе передаётся параметр preAuth, который имеет значение true (истина).
7	Платежный шлюз расшифровывает данные платежа и производит оплату.	
8	Платежный шлюз возвращает в приложение результат оплаты, который отображается на экране мобильного устройства/перенаправляет клиента на страницу результата оплаты.	
9	Продавец запрашивает у платежного шлюза статус оплаты заказа, используя уникальный идентификатор, полученный при регистрации заказа в параметре orderId: • getOrderStatusExtended.do.	
10	Не актуально	Для списания средств со счёта клиента продавец должен направить в платёжный шлюз запрос завершения оплаты: • deposit.do.
11	Не актуально	Платежный шлюз возвращает результат обработки запроса. Статус заказа не возвращается. Для получения статуса заказа необходимо отправить в шлюз запрос состояния заказа: • [integration:api:rest:requests:getorderstatusextended_cart getOrderStatusExtended.do]].
12	Если клиент получил товар или услугу не в момент оплаты, необходимо отправить запрос на создание чека закрытия .	

Mir Pay (Расшифровка платежных данных на стороне Мерчанта)

При оплате с использованием Mir Pay, где расшифровка платежных данных происходит на стороне мерчанта, взаимодействие происходит по следующей схеме.

	Одностадийный платёж	Двухстадийный платёж
1	Пользователь выбирает способ оплаты Mir Pay.	
2	Мерчант через MirPaySDK запускает сценарий подготовки In-Application операции в приложении Mir Pay/Мерчант через Deeplink или Universal Link запускает сценарий подготовки In-Application операции в приложении Mir Pay.	
3	Mir Pay отображает список карт клиенту.	
4	Клиент выбирает карту для оплаты в приложении MirPay.	
5	Mir Pay формирует данные платежа и отправляет зашифрованные данные платежа мерчанту.	
6	Мерчант расшифровывает данные платежа.	

	Одностадийный платёж	Двухстадийный платёж
7	Мерчант отправляет в платежный шлюз запрос на оплату, передавая расшифрованные данные платежа: • /mir/paymentDirect.do. При этом: • параметр preAuth либо не передается, либо имеет значение false (ложь).	Мерчант отправляет в платежный шлюз запрос на оплату: • /mir/paymentDirect.do; При этом: • в запросе передается параметр preAuth, который имеет значение true (истина).
8	Платежный шлюз производит оплату.	
9	Платежный шлюз возвращает в приложение результат оплаты, который отображается на экране мобильного устройства/перенаправляет клиента на страницу результата оплаты.	
10	Продавец запрашивает у платежного шлюза статус оплаты заказа, используя уникальный идентификатор, полученный при регистрации заказа в параметре orderId: • getOrderStatusExtended.do.	
11	Не актуально	Для списания средств со счёта клиента продавец должен направить в платёжный шлюз запрос завершения оплаты: • deposit.do.
12	Не актуально	Платежный шлюз возвращает результат обработки запроса. Статус заказа не возвращается. Для получения статуса заказа необходимо отправить в шлюз запрос состояния заказа: • getOrderStatusExtended.do.
13	Если клиент получил товар или услугу не в момент оплаты, необходимо отправить запрос на создание чека закрытия .	

Оплата с предоставлением возможности интернет-кредитования

Схема взаимодействий при использовании интернет-кредитования

	Одностадийная оплата	Двухстадийная оплата
1	Клиент сформировал заказ на сайте продавца и выбрал функцию «Оплата в кредит».	
2	Система продавца регистрирует заказ в платёжном шлюзе: * запрос на регистрацию заказа или * запрос на регистрацию заказа с фискализацией	Система продавца отправляет в платёжный шлюз: * запрос на регистрацию заказа с предавторизацией или * запрос на регистрацию заказа с предавторизацией (фискализация)
3	Платёжный шлюз передаёт продавцу результат регистрации заказа, включая уникальный идентификатор заказа и URL страницы выбора срока кредитования, на который надо перенаправить браузер Клиента.	
4	Продавец перенаправляет браузер клиента на URL, полученный на предыдущем шаге.	
5	Браузер клиента переходит по указанному URL.	
6	Клиент получает страницу с описанием услуги кредитования.	
7	Клиент нажимает кнопку Подать заявку .	
8	Платёжный шлюз регистрирует заявку.	
9	Платёжный шлюз перенаправляет браузер клиента на URL Сбербанк Онлайн.	

	Одностадийная оплата	Двухстадийная оплата
10	Браузер клиента переходит на страницу аутентификации Сбербанк Онлайн.	
11	В личном кабинете Сбербанк Онлайн клиент выбирает срок кредитования и проходит процедуру выдачи кредита.	
12	Сбербанк Онлайн обменивается данными с платёжным шлюзом.	
13	Сбербанк Онлайн производит оплату.	Заявка на кредит находится в статусе Одобен и ожидает завершения со стороны партнера в течение 7 суток с момента одобрения. Если в течение этого времени со стороны партнера не поступит запрос завершения заказа <code>deposit.do</code> , то заявка отклонится, заказ будет отменен.
14	Сбербанк Онлайн информирует РБС об изменении статуса заказа.	
15	После проведения оплаты Сбербанк Онлайн перенаправляет клиента на страницу продавца.	
16	Платёжный шлюз обменивается данными с ОФД.	
17	Платёжный шлюз производит обновление статуса заказа.	
18	Продавец направляет в платёжный шлюз запрос статуса заказа .	
19	Платёжный шлюз возвращает ответ на запрос статуса заказа.	
20	Не актуально	Для списания средств со счёта клиента продавец должен направить в платёжный шлюз запрос завершения оплаты: • завершение с оплатой полной суммой: ◦ deposit.do; • завершение с оплатой части суммы: ◦ deposit.do.
21	Не актуально	Платёжный шлюз возвращает результат обработки запроса. Статус заказа не возвращается. Для получения статуса заказа необходимо отправить в шлюз запрос состояния заказа: • getOrderStatusExtended.do.

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса заказа.

Отмена платежа

Отмена оплаты заказа доступна при наличии соответствующих прав по согласованию со «Сбербанком».

- При одностадийных платежах отмена платежа возможна для заказов в состоянии **Завершён/Deposited**, но только до закрытия дня, т. е. до 24:00 (по московскому времени) того дня, когда была произведена оплата.
- При двухстадийных платежах отмену платежа можно выполнить для заказа в состоянии

Подтверждён/Approved.

В таблице ниже представлена схема взаимодействий продавца и платёжного шлюза при отмене оплаты.

1	Продавец делает запрос на отмену платежа: • reverse.do.
2	После получения успешного ответа платёжного шлюза продавец делает запрос статуса заказа: • Расширенный запрос состояния заказа (getOrderStatusExtended.do);

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса заказа.

Возврат средств плательщику

Платёжный шлюз позволяет осуществлять полный или частичный возврат средств по оплаченным заказам.

Возврат средств возможен для платежей в состоянии **Завершён/Deposited**.

В таблице ниже представлена схема взаимодействий продавца и платёжного шлюза при возврате средств покупателю.

1	Продавец делает запрос на возврат средств покупателю: • возврат полной суммы: ◦ refund.do; • частичный возврат: ◦ refund.do.
2	После получения успешного ответа платёжного шлюза продавец делает запрос статуса заказа: • getOrderStatusExtended.do.
3	Если для заказа был осуществлен частичный возврат , то продавец делает запрос закрытия чека: • closeOfdReceipt.do.

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса

заказа.

Оплата по связкам

Оплата по связке (платёжная форма размещена на сервере платёжного шлюза)

	Одностадийная оплата	Двухстадийная оплата
1	Платательщик выбирает товар или услугу на ресурсе продавца и выбирает способ оплаты банковской картой.	
2	Продавец отправляет в платёжный шлюз запрос на регистрацию заказа: • register.do. В том числе в запросе передаются следующие параметры: • <code>clientId</code> - идентификатор клиента в системе магазина; • <code>returnUrl</code> - URL-адрес, на который будет перенаправлен платательщик в случае успешного платежа; • <code>failUrl</code> - URL-адрес, на который будет перенаправлен платательщик в случае неуспешного платежа. При первичной передаче параметра <code>clientId</code> происходит регистрация идентификатора клиента в платёжном шлюзе и сохранение данных его карты (кроме CVC-кода). При последующих передачах этого параметра клиент перенаправляется на страницу, где ему предлагается выбор: • использовать ранее сохранённые данные платёжной карты - в этом случае клиенту не нужно повторно заполнять данные платёжной карты для оплаты; • ввести данные новой платёжной карты - в этом случае к идентификатору этого клиента будут дополнительно привязаны данные новой платёжной карты.	Продавец отправляет в платёжный шлюз запрос на регистрацию заказа с предавторизацией: • registerPreAuth.do. В том числе в запросе передаются следующие параметры: • <code>clientId</code> - идентификатор клиента в системе магазина; • <code>amount</code> - сумма списания; • <code>orderNumber</code> - номер заказа в системе магазина; • <code>returnUrl</code> - URL-адрес, на который будет перенаправлен платательщик в случае успешного платежа; • <code>failUrl</code> - URL-адрес, на который будет перенаправлен платательщик в случае неуспешного платежа. При первичной передаче параметра <code>clientId</code> происходит регистрация идентификатора клиента в платёжном шлюзе и сохранение данных его карты (кроме CVC-кода). При последующих передачах этого параметра клиент перенаправляется на страницу, где ему предлагается выбор: • использовать ранее сохранённые данные платёжной карты - в этом случае клиенту не нужно повторно заполнять данные платёжной карты для оплаты; • ввести данные новой платёжной карты - в этом случае к идентификатору этого клиента будут дополнительно привязаны данные новой платёжной карты.
3	Платёжный шлюз в ответе в числе прочего возвращает параметры: • <code>orderId</code> - уникальный идентификатор заказа в платёжной системе; • <code>formUrl</code> - URL-адрес формы ввода данных банковской карты (платёжная форма).	

	Одностадийная оплата	Двухстадийная оплата
4	Продавец перенаправляет плательщику URL-адрес, указанный в параметре returnUrl ответа.	
5	В браузере плательщика отображается форма ввода данных банковской карты. Плательщик заполняет форму и отправляет данные на сервер платёжного шлюза.	
6	Дальнейшие действия зависят от того, поддерживает ли карта покупателя 3-D Secure: - поддержка 3-D Secure отсутствует - переход к следующему шагу процедуры; - есть поддержка 3-D Secure - платёжный шлюз перенаправляет плательщика на форму аутентификации (чаще всего это проверка с помощью SMS-сообщения), расположенную на сервере контроля доступа, принадлежащего банку-эмитенту - в случае успешной аутентификации выполняется следующий шаг сценария.	
7	Платёжный шлюз списывает деньги со счёта плательщика.	Платёжный шлюз производит холдирование (удержание) средств на счёте плательщика.
8	После проведения оплаты платёжный шлюз перенаправляет клиента на URL-адрес возврата (указанный магазином в запросе на регистрацию).	
9	Браузер клиента запрашивает страницу с результатами оплаты у магазина.	
10	Продавец запрашивает у платёжного шлюза статус оплаты заказа, используя уникальный идентификатор, полученный при регистрации заказа в параметре orderId: getOrderStatusExtended.do.	
11	Платёжный шлюз возвращает статус оплаты и продавец передаёт в браузер клиента страницу с результатами оплаты.	
12	Не актуально	Для списания средств со счёта клиента продавец должен направить в платёжный шлюз запрос завершения оплаты: deposit.do.
13	Не актуально	Платёжный шлюз возвращает результат обработки запроса. Статус заказа не возвращается. Для получения статуса заказа необходимо отправить в шлюз запрос состояния заказа: getOrderStatusExtended.do.

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса заказа.

Отмена платежа

Отмена оплаты заказа доступна при наличии соответствующих прав по согласованию со «Сбербанком».

- При одностадийных платежах отмена платежа возможна для заказов в состоянии **Завершён/Deposited**, но только до закрытия дня, т. е. до 24:00 (по московскому времени) того дня, когда была произведена оплата.
- При двухстадийных платежах отмену платежа можно выполнить для заказа в состоянии

Подтверждён/Approved.

В таблице ниже представлена схема взаимодействий продавца и платёжного шлюза при отмене оплаты.

1	Продавец делает запрос на отмену платежа: • reverse.do.
2	После получения успешного ответа платёжного шлюза продавец делает запрос статуса заказа: • Расширенный запрос состояния заказа (getOrderStatusExtended.do);

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса заказа.

Возврат средств плательщику

Платёжный шлюз позволяет осуществлять полный или частичный возврат средств по оплаченным заказам.

Возврат средств возможен для платежей в состоянии **Завершён/Deposited**.

В таблице ниже представлена схема взаимодействий продавца и платёжного шлюза при возврате средств покупателю.

1	Продавец делает запрос на возврат средств покупателю: • возврат полной суммы: ◦ refund.do; • частичный возврат: ◦ refund.do.
2	После получения успешного ответа платёжного шлюза продавец делает запрос статуса заказа: • getOrderStatusExtended.do.
3	Если для заказа был осуществлен частичный возврат , то продавец делает запрос закрытия чека: • closeOfdReceipt.do.

Для получения информации о состоянии заказа вместо отправки запроса `getOrderStatusExtended` вы можете получать [уведомления обратного вызова](#). Эти уведомления будут отправляться вам автоматически, в том числе после изменения статуса

заказа.

Операции со связками

- [Запрос деактивации связки](#)
- [Запрос активации ранее деактивированной связки](#)
- [Запрос списка связок по идентификатору клиента](#)
- [Запрос списка связок определённой банковской карты](#)
- [Запрос изменения срока действия связки](#)
- [Запрос создания связки без проведения платежа](#)

```
{(rater>id=rater8_1|name=Была ли статья полезна?|type=vote)}
```