

Платёжный шлюз «ВсеПлатежи»

Руководство Мерчанта

Оглавление

История изменений.....	3
Раздел 1 Определения.....	6
Раздел 2 Общие сведения.....	8
2.1 Защищённое соединение.....	8
2.2 Способы проведения оплаты.....	9
2.3 Алгоритм проведения оплаты.....	10
2.3.1 Уведомление об успешном проведении оплаты.....	14
2.3.2 Уведомление об отклонении операции по SberPay.....	15
Раздел 3 Платёжная форма.....	16
3.1 Общий вид.....	16
3.2 Основные элементы формы.....	17
Раздел 4 API.....	19
4.1 Адреса серверов.....	19
4.2 Поддерживаемые запросы.....	19
4.2.1 Запрос на проведение оплаты.....	22
4.2.2 Запрос статуса заказа.....	23
4.2.3 Запрос расширенного статуса заказа.....	25
4.2.4 Запрос проведение рекуррентного платежа.....	27
4.2.5 Запрос на передачу данных для информирования ГИС ЖКХ.....	28
4.2.6 Запрос на блокировку средств.....	29
4.2.7 Запрос на списание заблокированных средств.....	30
4.2.8 Запрос на разблокировку средств.....	31
4.2.9 Запрос на привязку карты.....	32
4.2.10 Запрос на получение списка сохранённых карт.....	32
4.2.11 Запрос для сохранения карты.....	33
4.2.12 Запрос для удаления карты.....	34
4.2.13 Запрос для оплаты дополнительными способами через API.....	34
4.2.14 Запрос доступности дополнительных способов оплаты через API.....	38
4.2.15 Запрос отмены оплаты (SberPay).....	39
4.2.16 Коды и тексты состояний оплаты.....	40
4.2.17 Коды и тексты состояний транзакций.....	41
4.2.18 Коды ответов на запросы по API.....	41
4.3 Алгоритм формирования НМАС.....	43
4.3.1 Подготовка строки данных для НМАС.....	43
4.3.2 Генерация НМАС.....	44
4.4 Примеры для некоторых языков программирования.....	45
4.4.1 PHP.....	45
4.4.2 Java.....	45
4.5 Дополнительные параметры в запросы по API.....	45
Раздел 5 Платёжный виджет.....	49
5.1 Общее описание.....	49
5.2 Подключение виджета.....	49
5.3 Параметры вызова.....	51
Раздел 6 Реестр проведённых операций.....	52
6.1 Формирование и отправка реестра.....	52
6.2 Формат письма.....	52
6.3 Формат файла реестра.....	52
Раздел 7 Платёжная ссылка.....	54
7.1 Формирование и отправка ссылки для оплаты из личного кабинета.....	54

История изменений

Версия	Дата	Изменения
3.0	14.03.2017	Документ создан на основе переработанной версии 2.0.
3.1	26.05.2017	- Добавлена история изменений. - Добавлен п. 2.1 «Защищённое соединение». - Добавлено описание email-уведомлений (п. 2.2.1 «Уведомление об успешном проведении оплаты») - Актуализирован список кодов ответов на запрос статуса заказа (п. 4.2.2).
3.2	27.06.2017	В связи с добавлением поддержки платёжного виджета: - Дополнен Раздел 1 «Определения». - Переписан Раздел 2 «Общие сведения». - Добавлены отдельные разделы для платёжной формы и платёжного виджета.
3.3	22.08.2017	- Обновлён Раздел 1 «Определения». - Добавлена графическая схема оплаты в Раздел 2 «Общие сведения». - Добавлен пункт «Основные элементы формы» в Раздел 3 «Платёжная форма». - Добавлен пункт «Параметры вызова» в Раздел 5 «Платёжный виджет». - Добавлен Раздел 6 «Реестр проведённых операций».
3.4	02.11.2017	- Обновлён пункт 2.3.1 «Уведомление об успешном проведении оплаты». - Добавлен пункт 4.2.3 «Запрос расширенного статуса заказа».
3.5	03.12.2018	Добавлена возможность создания и проведения рекуррентных платежей (автоплатежи): - Обновлён пункт 4.2.1 «Запрос на проведение платежа» - Обновлён пункт 4.2.2 «Запрос статуса заказа» - Добавлен пункт 4.2.4 «Запрос на проведение рекуррентного платежа»
3.6	25.12.2018	Добавлен запрос для передачи данных, необходимых при информировании ГИС ЖКХ: - Обновлён пункт 4.2 «Поддерживаемые запросы» - Добавлен пункт 4.2.5 «Запрос на передачу данных для информирования ГИС ЖКХ»
3.7	24.06.2019	Добавлен параметр для передачи номера телефона плательщика. Обновлены разделы: 3.1 Общий вид 2.3.1 Уведомление об успешном проведении оплаты 4.2.1 Запрос на проведение оплаты 4.2.2 Запрос статуса заказа 4.2.3 Запрос расширенного статуса заказа 4.2.4 Запрос проведение рекуррентного платежа
3.8	22.01.2020	Обновлены разделы:

		4.2.5 Запрос на передачу данных для информирования ГИС ЖКХ 4.3.1 Подготовка строки данных для НМАС
3.9	26.05.2020	Обновлён раздел 4.2 Поддерживаемые запросы; 4.2.6 Запрос на блокировку средств 4.2.7 Запрос на списание заблокированных средств 4.2.8 Запрос на разблокировку средств
4.0	14.10.2020	Обновлён раздел 4.2.4 Запрос проведение рекуррентного платежа
4.1	05.02.2021	Обновлён раздел 4.2 Поддерживаемые запросы; 4.2.6 Запрос на блокировку средств 4.2.7 Запрос на списание заблокированных средств 4.2.8 Запрос на разблокировку средств
4.2	22.03.2021	Обновлён раздел 4.2.1 Запрос на проведение оплаты, 4.2.6 Запрос на блокировку средств
4.3	11.06.2021	Добавлен раздел 4.2.9 Запрос на привязку карты
4.4	23.06.2021	Обновлён раздел: 4.2.3 Запрос расширенного статуса заказа Добавлен раздел: 4.2.11 Коды и тексты состояний транзакций
4.5	12.07.2021	Обновлены разделы: 4.2 Поддерживаемые запросы 4.2.1 Запрос на проведение оплаты 4.2.3 Запрос расширенного статуса заказа 4.2.6 Запрос на блокировку средств Добавлены разделы: 4.2.10 Запрос на получение списка сохранённых карт 4.2.11 Запрос для сохранения карты 4.2.12 Запрос для удаления карты
4.6	10.01.2022	Обновлён раздел: 4.2.10 Запрос на получение списка сохранённых карт
4.7	27.01.2022	Добавлены разделы: 4.2.13 Запрос для оплаты дополнительными способами через API 4.2.14 Запрос доступности дополнительных способов оплаты через API
4.8	14.04.2022	Обновлены разделы: 4.2.13 Запрос для оплаты дополнительными способами через API 4.2.14 Запрос доступности дополнительных способов оплаты через API
4.9	16.05.2022	Добавлены разделы: 2.3.2 Уведомление об отклонении операции по SberPay 4.2.15 Запрос отмены оплаты (SberPay)
5.0	31.05.2022	Добавлен способ оплаты YandexPay

5.1	05.10.2022	Добавлен Раздел 7 Платёжная ссылка
5.2	27.12.2022	Обновлены разделы: Раздел 1 Определения 4.2.13 Запрос для оплаты дополнительными способами через API
5.3	12.08.2023	Обновлены разделы: 4.2.13 Запрос для оплаты дополнительными способами через API; 4.2.14 Запрос доступности дополнительных способов оплаты через API
5.4	14.08.2023	4.2.4 Запрос проведение рекуррентного платежа
5.5	18.09.2023	Обновлен раздел: Способы проведения оплаты
5.6	13.11.2023	Обновлен раздел: 4.2.13 Запрос для оплаты дополнительными способами через API
5.7	05.02.2024	Добавлен раздел: Коды ответов на запросы по API
5.8	05.03.2024	Обновлены разделы: Запрос на проведение оплаты Запрос проведение рекуррентного платежа Запрос на блокировку средств Запрос на списание заблокированных средств Запрос на привязку карты Запрос для оплаты дополнительными способами через API
5.9	12.03.2024	Добавлен раздел: 4.5 Дополнительные параметры в запросы по API

Раздел 1 Определения

Платёжный шлюз «ВсеПлатежи» — совокупность программных и аппаратных средств, выполняющих: 1) обработку запросов Торговых точек на проведение операций, 2) проведение операций, 3) передачу результатов проведения операций Торговым точкам. Далее используется термин «Платёжный шлюз».

Торговая точка (Мерчант) — клиент Платёжного шлюза, с которым заключён договор на оказание услуг Платёжного шлюза и налажено техническое взаимодействие.

Пользователь — лицо, инициирующее проведение операции оплаты, клиент Торговой точки.

Эквайринг (от англ. acquire — приобретать, получать) — приём к оплате платёжных карт в качестве средства оплаты товара, работ, услуг. (*Википедия*)

Интернет-эквайринг — это технология, являющаяся разновидностью эквайринга, позволяющая принимать к оплате банковские карты через Интернет. (*Википедия*)

Банк-эквайер — банк, уполномоченный принимать к оплате платёжные карты, посредством POS-терминалов или через интернет.

Банк-эмитент — банк, выпускающий в обращение банковские карты. Банк-эквайер выполняет запросы авторизации и списания денежных средств в банк-эмитент, т. е. в банк, выпустивший конкретную платёжную карту.

Фискальный чек — чек, который продавец формирует на онлайн-кассе для передачи покупателю или клиенту. 54-ФЗ обязывает продавцов выдавать покупателям и клиентам фискальные чеки. По запросу Мерчанта на платёжном шлюзе может быть настроена возможность сбора и передачи данных в онлайн-кассе партнёра для отправки чеков клиентам.

3-D Secure (3DS) — технология, которая используется как дополнительный уровень безопасности для онлайн-кредитных и дебетовых карт, двухфакторной аутентификации пользователя. 3-D Secure добавляет ещё один шаг аутентификации для онлайн-платежей, позволяющий торговым точкам и банкам дополнительно убедиться, что платёж совершает именно держатель карты, чтобы защититься от мошеннических операций. Обычно для такой аутентификации используется sms-пароль, отправляемый на привязанный к карте номер сотового телефона.

API — программный интерфейс для взаимодействия с каким-либо приложением или системой, в частности, с сервером Платёжного шлюза.

Виджет — элемент интерфейса, в частности, платёжная форма, встраиваемая на сайт Торговой точки.

Общий секретный ключ — набор случайных цифр в шестнадцатеричном формате, сгенерированный модулем безопасности Платёжного шлюза для формирования подписи

(HMAC). Ключ присваивается терминалу Торговой точки и должен храниться в тайне у обеих сторон.

HMAC — *hash-based message authenticate code (код аутентификации сообщений)* — набор символов, сформированный при обработке входящих параметров по алгоритму SHA256 с использованием общего секретного ключа. HMAC передаётся отдельным параметром `sign` в запросах от Торговой точки к API Платёжного шлюза и передаче ответов на эти запросы обратно Торговой точке. Предназначен для обеспечения целостности запроса и обоюдной аутентификации Платёжного шлюза и Торговой точки.

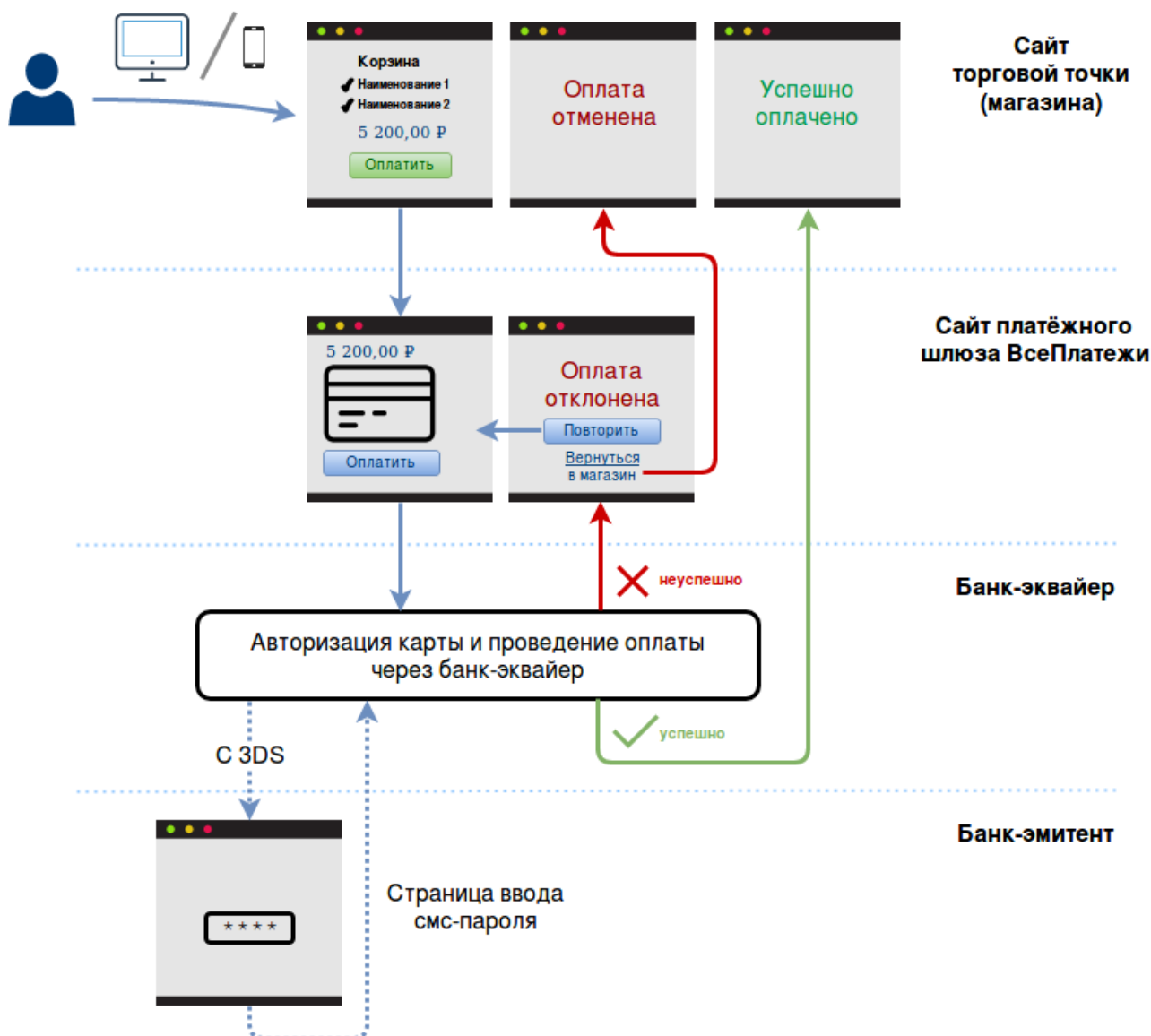


Рисунок **Ошибка! Только основной документ.**: Общая схема приёма платежа через API

Раздел 2 Общие сведения

2.1 Защищённое соединение

Все взаимодействия с платёжным шлюзом производятся по протоколу HTTPS. Для защиты передаваемой информации используется протокол TLS версии 1.2. Протоколы SSL всех версий и TLS версий ниже 1.2 не поддерживаются.

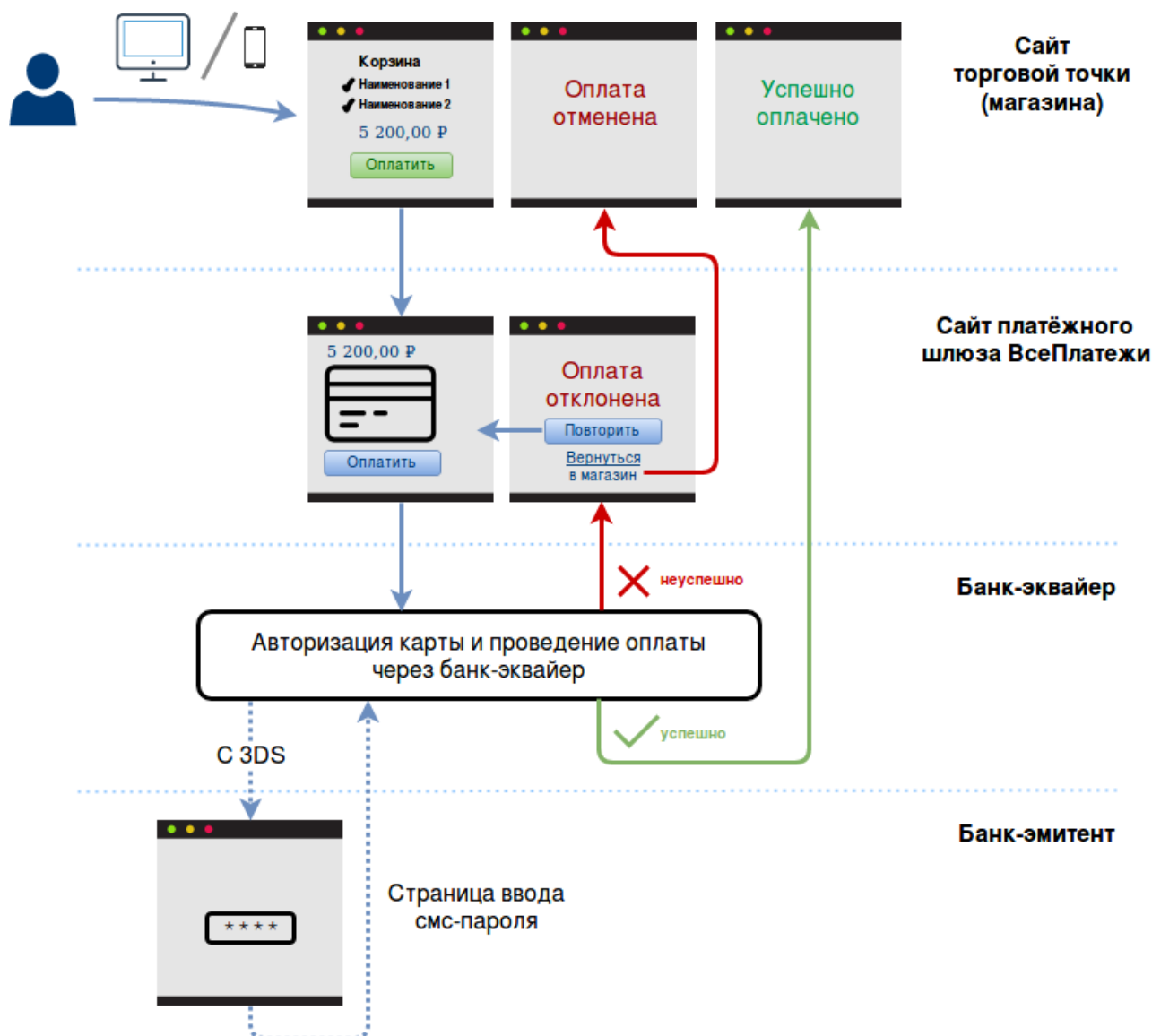


Рисунок **Ошибка! Только основной документ.**: Общая схема приёма платежа через API

2.2 Способы проведения оплаты

Платёжный шлюз поддерживает два способа интеграции: непосредственное взаимодействие через API или встраивание виджета с формой оплаты. Первый способ более надёжный, так как включает аутентификацию запросов через подпись, а также поддерживает передачу параметра `userid`, который даёт возможность пользователю привязывать введённую карту к своему условному идентификатору, устанавливая на форме оплаты галочку «Запомнить карту для будущих операций». В то же время, этот способ более затратный с технической точки зрения. Более простой вариант подключения – встраивание платёжного виджета на сайте Торговой точки, который требует только внедрения HTML и JavaScript-кода, генерировать подпись при этом не нужно, также некоторые другие параметры становятся необязательными.

У платёжного шлюза есть отдельные кнопки быстрых оплат. Для корректности работы в мобильном приложении Торговой точки должна быть реализована возможность перехода по `deepLink` (открытие ссылки в браузере по умолчанию на устройстве клиента).

Подробнее смотрите в Раздел 4 и Раздел 5.

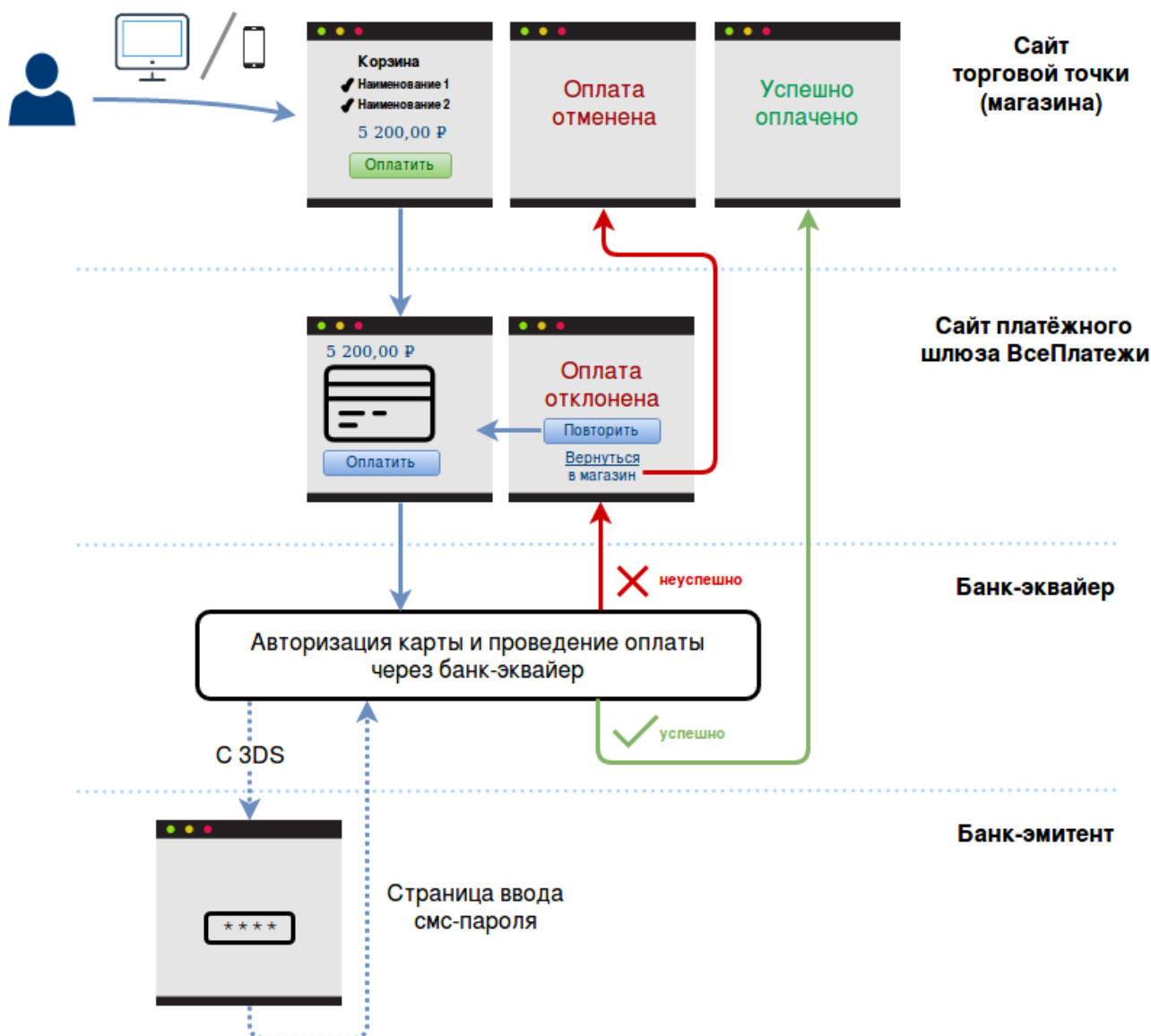


Рисунок **Ошибка! Только основной документ.**: Общая схема приёма платежа через API

2.3 Алгоритм проведения оплаты

Смотрите общую схему на Рисунке Рисунок 1. Пользователь инициирует оплату товара или услуги на стороне Торговой точки. В случае использования API Торговая точка формирует запрос с подписью и отправляет его как HTML-форму с непосредственным перенаправлением Пользователя на сайт Платёжного шлюза. При данном запросе шлюз проверяет параметры запроса и отображает форму оплаты, на которой Пользователь вводит реквизиты платёжной карты (номер карты, месяц/год действия карты, трёхзначный код безопасности (CVV2/CVC2)). Если запрос от Торговой точки сформирован неверно, то пользователь увидит страницу с ошибкой «Операция отклонена». В случае использования виджета платёжная форма открывается в модальном блоке на сайте Торговой точки. В обоих вариантах Пользователь заполняет поля формы и нажимает кнопку «Оплатить». После успешной верификации введённых Пользователем данных выполняются следующие шаги:

1. В случае успешного проведения операции Платёжный шлюз асинхронно (без блокировки пользовательского интерфейса) отправляет Торговой точке уведомление (см. п. 2.3.1).
2. Пользователь перенаправляется на страницу сайта Торговой точки по адресу, указанному в

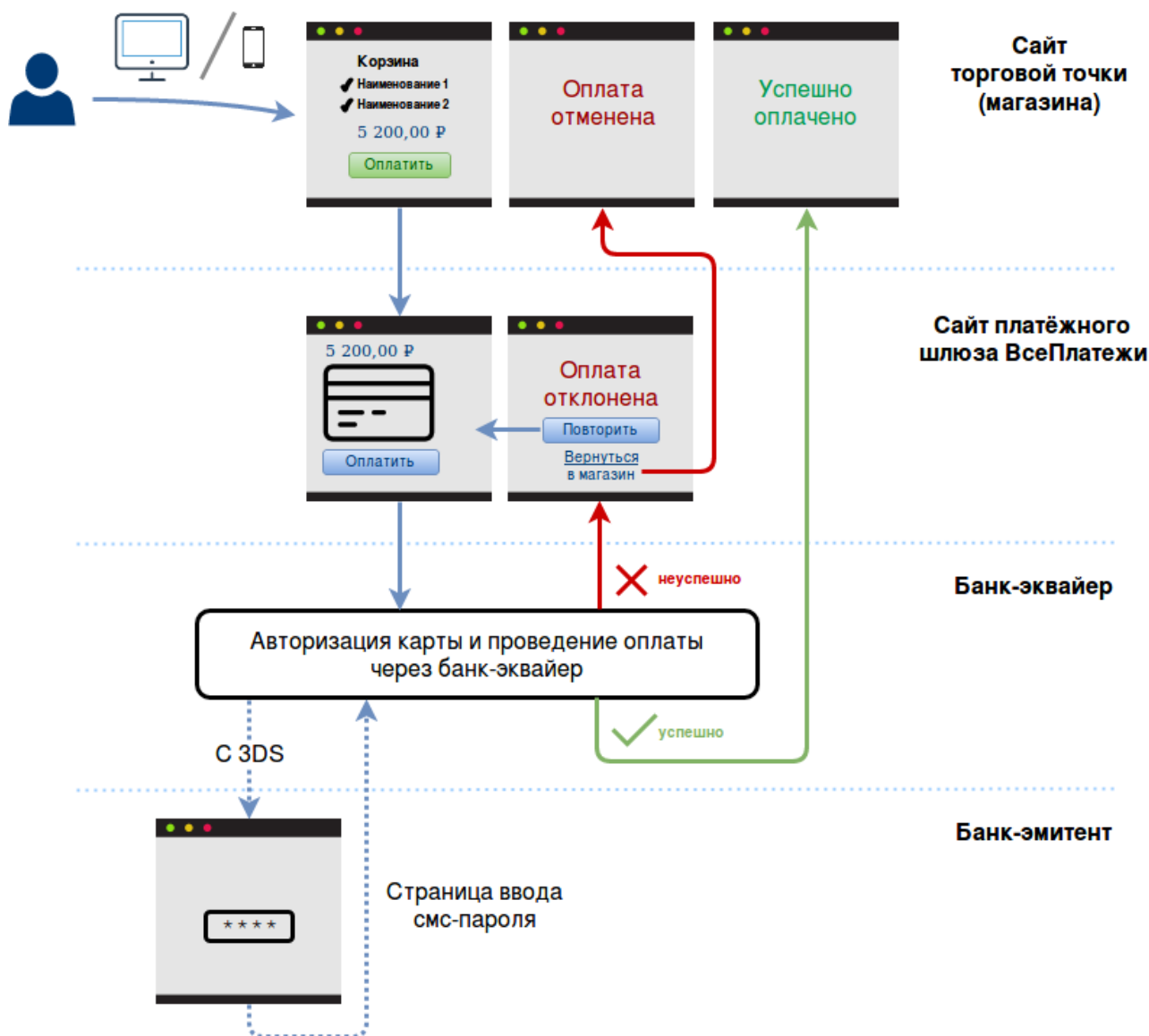


Рисунок **Ошибка! Только основной документ.**: Общая схема приёма платежа через API

clientBackUrl, с передачей дополнительного параметра `result=0`, говорящего об успехе операции оплаты. В случае неудачной отправки уведомления оно будет повторно отправляться заданное в конфигурации Платёжного шлюза число раз (по умолчанию – 3 раза) через заданный таймаут (по умолчанию – 2 минуты).

- В случае отказа в проведении оплаты Пользователю отображается страница, где описана причина отказа. В этом случае Пользователь может повторить оплату, вернувшись по ссылке на форму оплаты, либо вернуться на сайт Торговой точки, кликнув соответствующую ссылку на форме оплаты (Пользователь будет перенаправлен на `clientBackUrl` с передачей дополнительного параметра `result`, **не равного 0**, говорящего об отказе от операции оплаты). В случае использования виджета параметр `clientBackUrl` является необязательным. По умолчанию Пользователю будет отображена специальная страница Платёжного шлюза с квитанцией или с описанием ошибки.

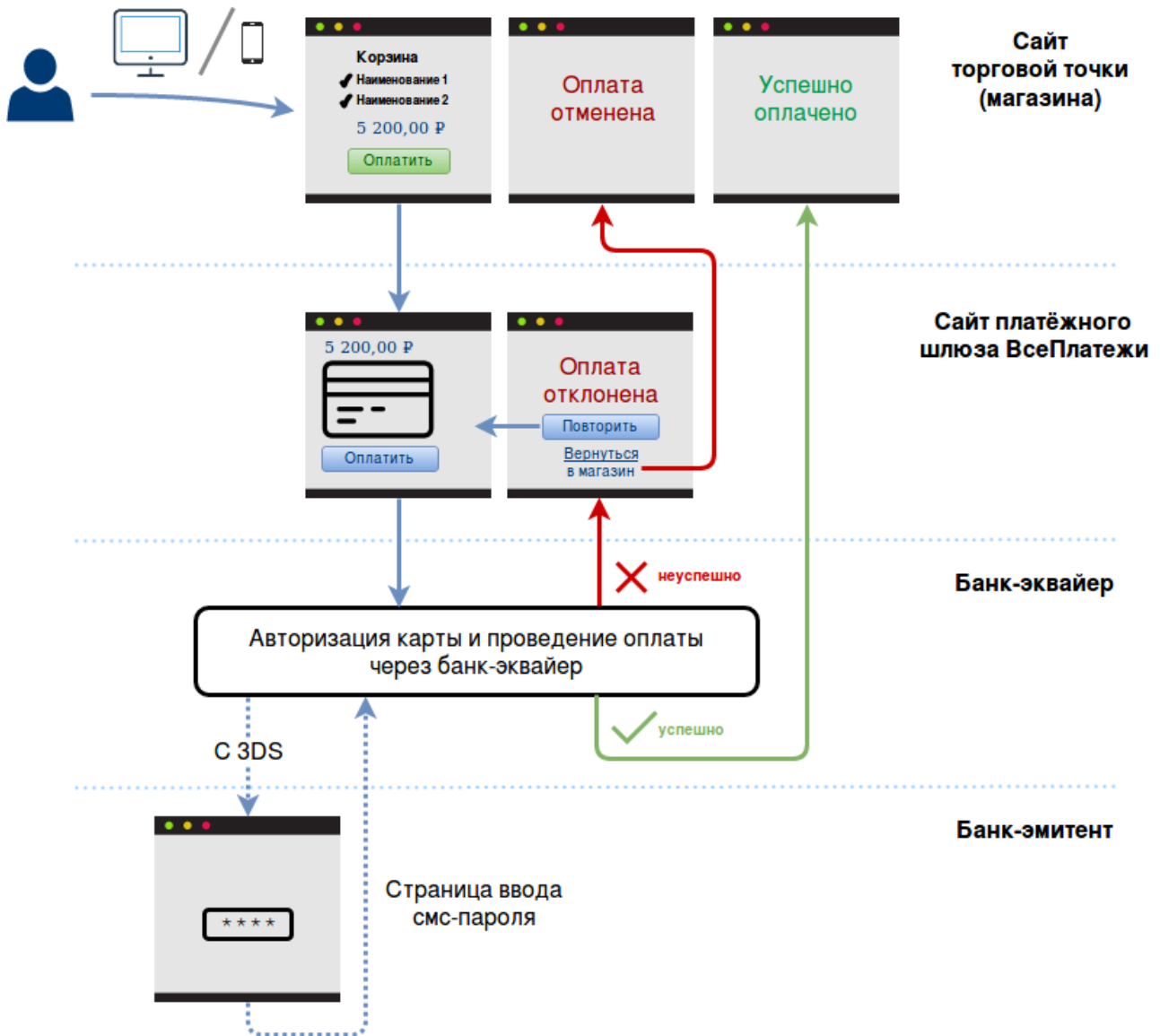
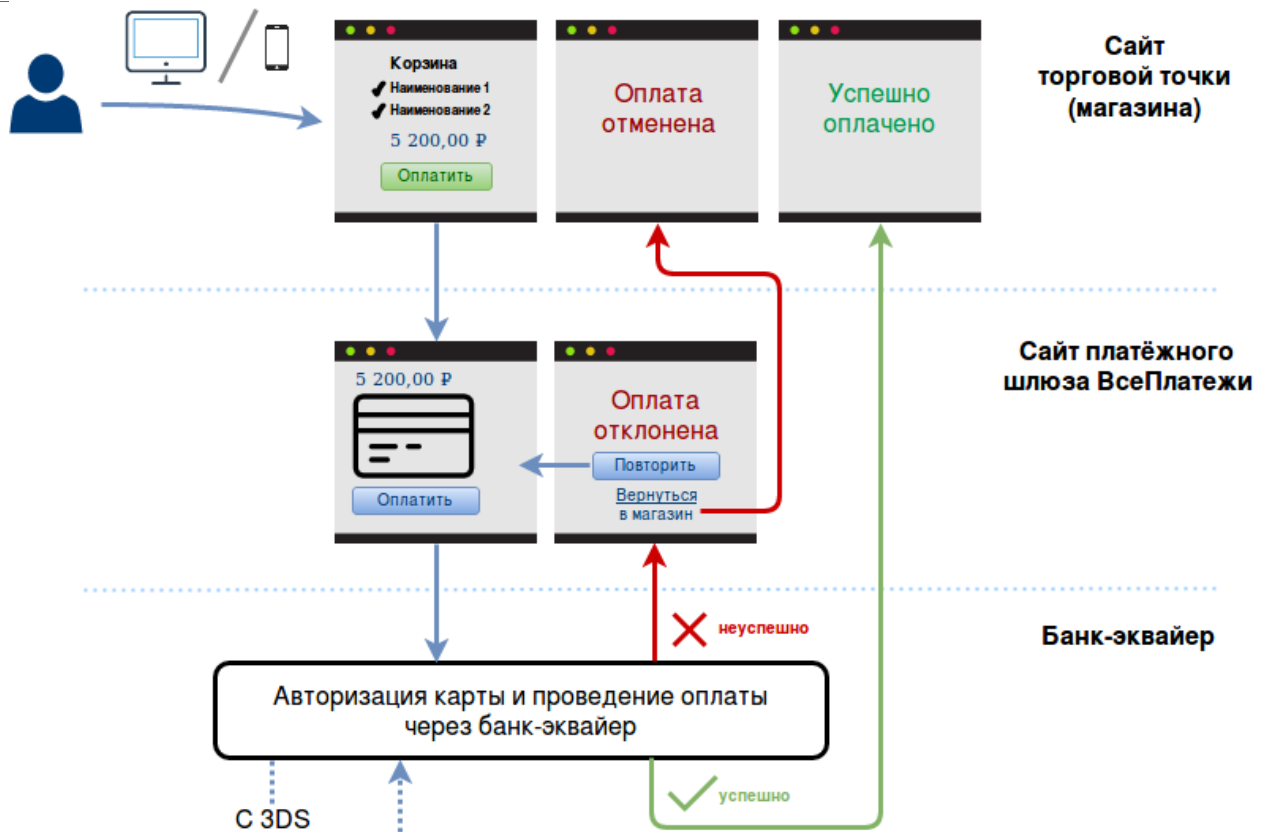


Рисунок **Ошибка! Только основной документ.**: Общая схема приёма платежа через API



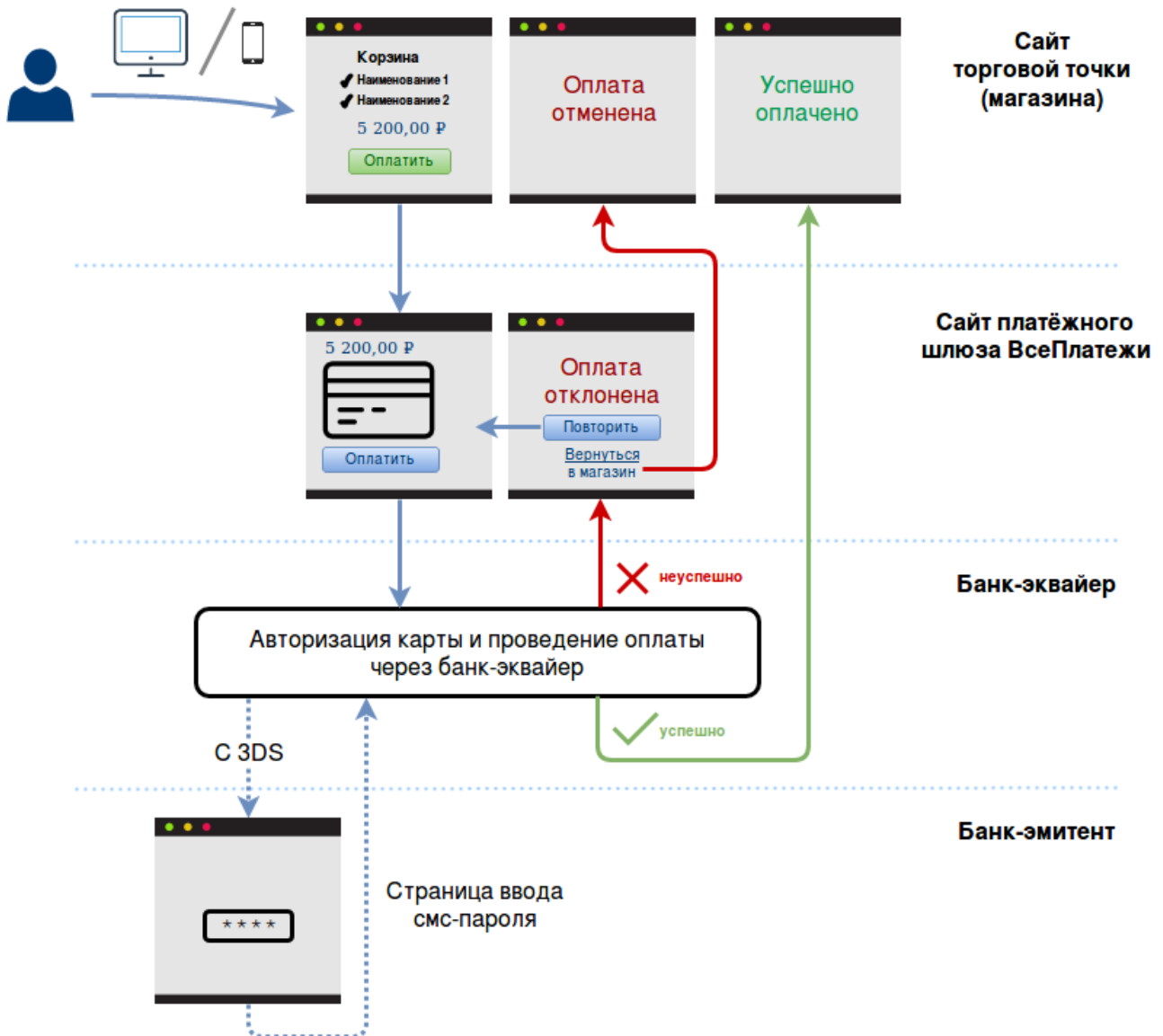


Рисунок **Ошибка! Только основной документ.**: Общая схема приёма платежа через API

Рисунок 1: Общая схема приёма платежа через API

2.3.1 Уведомление об успешном проведении оплаты

После успешной оплаты заказа, Платёжный шлюз асинхронно отправляет уведомление одним из двух способов: по протоколу HTTP или по электронной почте (email).

HTTP-уведомление

Отправляется в виде POST-запроса (с заголовком Content-Type: application/x-www-form-urlencoded) на согласованный заранее с Торговой точкой URL или на URL, переданный Торговой точкой по типу интеграции API (см. Раздел 4). Торговая точка сообщает выбранный вариант уведомлений для включения соответствующих настроек на стороне Платёжного шлюза.

Список отправляемых параметров:

1. orderId
2. amount
3. terminal
4. merchant
5. transactionId
6. transactionDateTime (в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС)
7. cardNumber
8. createdRecurrentTemplateId
9. email
10. phone
11. sign

Кроме параметра sign («подпись»), все остальные параметры берутся из иницирующего запроса на оплату, сформированного Торговой точкой. Подпись генерируется для указанных параметров по алгоритму, описанному в п. 4.2.18, и, в целях безопасности, должна проверяться на стороне Торговой точки.

Email-уведомление

Отправляется на согласованный заранее с Торговой точкой адрес или несколько адресов электронной почты. Пример сообщения:

Заказ <номер заказа> (<дата/время создания заказа>) на сумму <сумма заказа> успешно оплачен.

Описание заказа: <описание >

Терминал: <номер терминала> <описание терминала>

Номер карты: <номер карты в виде XXXXXX*****XXXX>

Транзакция: № <номер транзакции> (<дата/время перехода транзакции в статус «Оплачена»>)

Подробную информацию по заказу вы можете посмотреть в личном кабинете.

--

С уважением,

Служба поддержки клиентов «ВсеПлатежи».

2.3.2 Уведомление об отклонении операции по SberPay

В случае отклонения транзакции при оплате по SberPay Платёжный шлюз при необходимости может отправить HTTP-уведомление Торговой точке.

HTTP-уведомление

Отправляется в виде POST-запроса (с заголовком Content-Type: application/x-www-form-urlencoded) на переданный URL Торговой точкой по типу интеграции API (см. Раздел 4).

Список отправляемых параметров:

1. orderId
2. amount
3. terminal
4. merchant
5. transactionId
6. transactionDateTime (в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС)
7. transactionStatusCode
8. email
9. phone
10. ISO-код
11. sign

Подпись генерируется для указанных параметров по алгоритму, описанному в п. 4.2.18, и, в целях безопасности, должна проверяться на стороне Торговой точки.

Раздел 3 Платёжная форма

3.1 Общий вид

Платёжная форма имеет адаптивный интерфейс и подстраивается под ширину экрана или окна браузера на всех распространённых устройствах – от смартфонов до настольных компьютеров, примеры показаны на Рисунках 2 и 3. При необходимости платёжная форма может быть отображена в тёмном фоне при виде интеграции по API (см. Раздел 4).

The screenshot shows a payment form for 'ВсеПлатежи'. At the top left is the logo and name 'ВсеПлатежи'. At the top right are contact details: 'help@vp.ru' and '+7 800 700-08-38'. The main heading is 'Ввод данных для оплаты'. To the right of the heading is a timer: 'Успейте ввести данные за: 14:46'. Below the heading is a card input area with logos for VISA, mastercard, maestro, and MIR. The card number field contains '0000 0000 0000 0000'. Below the number is the expiration date field labeled 'Срок действия карты' with 'Месяц 00 / Год 00'. To the right of the expiration date is a field for 'Три цифры на обороте' containing '0 0 0'. Below the card input area is a blue button 'Оплатить 22.00 Р' and a link 'Отменить и вернуться'. At the bottom, there is a note: 'Нажимая на кнопку «Оплатить», вы соглашаетесь с условиями [оферты](#)'.

Рисунок 2: Вид платёжной формы на экране планшета или мониторе настольного компьютера

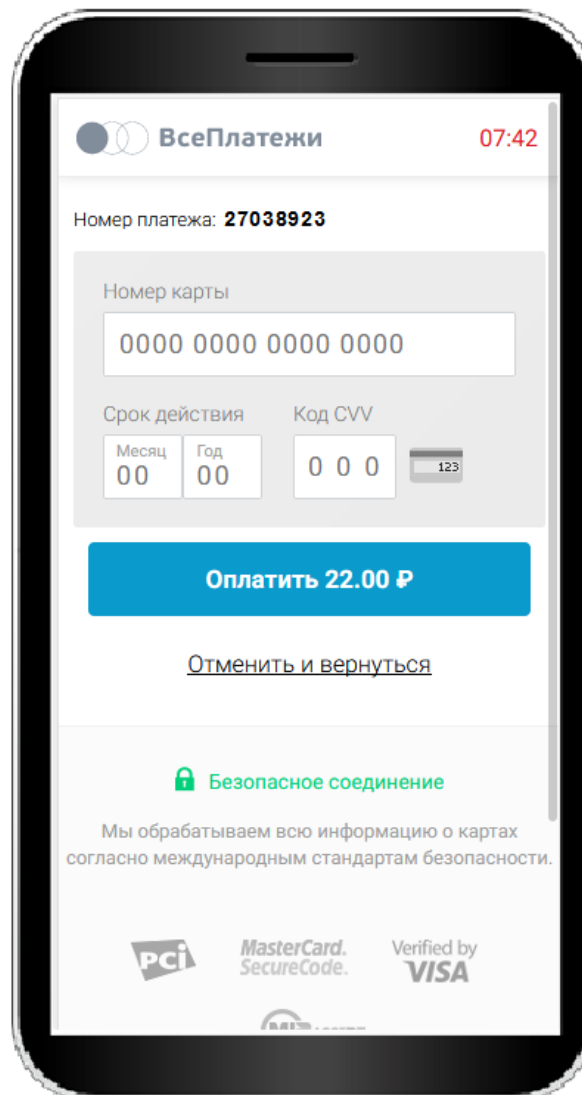


Рисунок 3: Вид платёжной формы на экране смартфона

3.2 Основные элементы формы

На форме отображается основная информация о платеже: сумма к оплате, номер платежа, который соответствует номеру заказа, и описание заказа, а также таймер времени, отведённого на оплату.

Если передан параметр `userid`, то отображается блок сохранённых карт. В нём пользователь может выбрать ранее сохранённую карту для оплаты или ввести данные новой карты. Карта будет привязана к переданному значению `userid`, если пользователь установит галочку «Запомнить карту для будущих действий». При этом не сохраняется код безопасности (CVV/CVC), поэтому при каждой оплате сохранённой картой пользователю необходимо вводить этот код. Номер карты хранится в зашифрованном виде и отображается только в маскированном виде (`****1234` или `123456*****1234`).

В зависимости от настроек терминала, может отображаться опция «Получить квитанцию», «Отправить фискальный чек»*, а также ссылка на оферту.

* На данный момент предоставляется возможность получения контакта пользователя для отправки фискального чека. Интеграции с электронными ККТ нет.

Раздел 4 API

4.1 Адреса серверов

Данные адреса необходимо использовать в качестве базовых, прибавляя к ним относительные URL, описанные далее.

Основной	https://gate.vp.ru
Тестовый	https://testgate.vseplatezhi.ru

4.2 Поддерживаемые запросы

В данной таблице в общем виде описаны поддерживаемые методы работы с Платёжным шлюзом.

Запрос	HTTP-метод и относительный URL	Результат
Запрос на проведение оплаты через API	POST /main	<p><i>Успешно:</i> Отображение страницы с формой оплаты (формой ввода данных карты).</p> <p><i>Ошибка:</i> Отображение страницы с описанием ошибки и HTTP-кодом, соответствующим типу ошибки.</p>
Запрос на проведение блокировки средств через API	POST /blockpage	<p><i>Успешно:</i> Отображение страницы с формой оплаты (формой ввода данных карты).</p> <p><i>Ошибка:</i> Отображение страницы с описанием ошибки и HTTP-кодом, соответствующим типу ошибки.</p>

<p>Запрос на проведение списания заблокированных средств через API</p>	<p>POST /charge</p>	<p><i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><i>Ошибка:</i> Ответ с соответствующим HTTP- кодом и JSON-строкой в теле сообщения.</p>
<p>Запрос на разблокировку средств через API</p>	<p>POST /retrieve</p>	<p><i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><i>Ошибка:</i> Ответ с соответствующим HTTP- кодом и JSON-строкой в теле сообщения.</p>
<p>Запрос для привязки карты через API</p>	<p>POST /attachpage</p>	<p><i>Успешно:</i> Отображение страницы с формой оплаты (формой ввода данных карты).</p> <p><i>Ошибка:</i> Отображение страницы с описанием ошибки и HTTP-кодом, соответствующим типу ошибки.</p>
<p>Запрос на проведение рекуррентного платежа через API</p>	<p>POST /recurrent</p>	<p><i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><i>Ошибка:</i> Пустой ответ с HTTP-кодом, соответствующим типу ошибки.</p>
<p>Запрос статуса заказа</p>	<p>POST /api/order/status</p>	<p><i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><i>Ошибка:</i> Пустой ответ с HTTP-кодом, соответствующим типу ошибки.</p>

Запрос расширенного статуса заказа	POST /api/order/status-ext	<p><i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><i>Ошибка:</i> Пустой ответ с HTTP-кодом, соответствующем типу ошибки.</p>
Запрос на передачу данных для информирования ГИС ЖКХ	POST /api/gis/	<p><i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><i>Ошибка:</i> Пустой ответ с HTTP-кодом, соответствующем типу ошибки.</p>
Запрос на получение списка сохранённых карт	GET /api/userid/cards	<p><i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><i>Ошибка:</i> Пустой ответ с HTTP-кодом, соответствующем типу ошибки.</p>
Запрос для сохранения карты	PUT /api/userid/card	<p><i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><i>Ошибка:</i> Пустой ответ с HTTP-кодом, соответствующем типу ошибки.</p>
Запрос для удаления карты	DELETE /api/userid/card	<p><i>Успешно:</i> Ответ с HTTP- кодом 204.</p> <p><i>Ошибка:</i> Пустой ответ с HTTP-кодом, соответствующем типу ошибки.</p>
Запрос для оплаты дополнительными способами через API	POST /api/token/payment/	<p><i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p>

		<i>Ошибка:</i> Ответ с соответствующим HTTP- кодом и JSON-строкой в теле сообщения.
Запрос доступности дополнительных способов оплаты через API	POST /api/token/payment/ activity/	<i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения. <i>Ошибка:</i> Ответ с соответствующим HTTP- кодом и JSON-строкой в теле сообщения.
Запрос отмены оплаты (Sber Pay)	POST /api/token/payment/decline/sber-pay/	<i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения. <i>Ошибка:</i> Ответ с соответствующим HTTP- кодом и JSON-строкой в теле сообщения.

4.2.1 Запрос на проведение оплаты

Базовые параметры

URL	/main
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
orderId	Номер заказа	Уникальный для терминала, числовой, 1-50
amount	Сумма заказа в рублях и копейках	Числовой, с двумя знаками после точки, > 0.00
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой,

		1-50
clientBackUrl	URL для возврата на сайт Торговой точки	Символьный, 1-255
recurrent	Флаг, определяющий необходимость создания рекуррентного платежа. По умолчанию создание отключено (соответствует отсутствию параметра или значению параметра false). При необходимости создания рекуррентного платежа в данном параметре необходимо передавать true	Логический, true/false Опциональный
sign	Подпись запроса	Символьный, 64 (для HMAC SHA256)

4.2.2 Запрос статуса заказа

Базовые параметры

URL	/api/order/status
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание
orderId	Номер заказа
merchant	Номер Торговой точки
terminal	Номер терминала Торговой точки
sign	HMAC запроса

Формат значений параметров смотрите в п. 4.2.1

Успешный ответ содержит JSON-структуру

Оплата заказа пользователем:

```
{
  "data": {
    "orderId": "<номер оплаты>",
    "amount": "<сумма оплаты>",
    "merchant": "<номер торговой точки>",
    "terminal": "<номер терминала>",
    "userId": "<ID пользователя мерчанта>"
  }
}
```

```
"orderStatusCode": "<код состояния оплаты>", -- см. секцию 4.2.13
"orderStatusText": "<название состояния оплаты>"
"recurrent": "true", -- при recurrent=true
"createRecurrentTemplateId": "<номер шаблона>"*,
"refunds": [] -- список проведённых возвратов
"email": "<адрес электронной почты>",
"phone": "<номер телефона>".
}
}
```

* при recurrent=true, если удалось создать шаблон платежа.

Рекуррентный платеж:

```
{
  "data": {
    "orderId": "<номер оплаты>",
    "amount": "<сумма оплаты>",
    "merchant": "<номер торговой точки>",
    "terminal": "<номер терминала>",
    "userId": "<ID пользователя мерчанта>",
    "orderStatusCode": "<код состояния оплаты>", -- см. секцию 4.2.13
    "orderStatusText": "<название состояния оплаты>"
    "recurrentTemplateId": "<номер шаблона>",
    "refunds": [] -- список проведенных возвратов
    "email": "<адрес электронной почты>",
    "phone": "<номер телефона>".
  }
}
```

4.2.3 Запрос расширенного статуса заказа

Базовые параметры

URL	/api/order/status-ext
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание
orderId	Номер заказа
Merchant	Номер Торговой точки
Terminal	Номер терминала Торговой точки
Sign	HMAC запроса

Формат значений параметров смотрите в п. 4.2.1

Успешный ответ содержит JSON-структуру

Оплата заказа пользователем:

```
{
  "data": {
    "orderId": "<номер оплаты>",
    "amount": "<сумма оплаты>",
    "merchant": "<номер торговой точки>",
    "terminal": "<номер терминала>",
    "userId": "<ID пользователя мерчанта>",
    "orderStatusCode": "<код состояния оплаты>", -- см. секцию 4.2.13
    "orderStatusText": "<название состояния оплаты>"
    "createdRecurrentTemplateId": "<номер шаблона рекуррентного платежа>"*
    "refunds": [ {
      "originalTransactionId": "<номер возвращенной транзакции>",
      "dateTime": "<дата проведения>",
      "amount": "<сумма возврата>"
    }, { ... } ] -- список проведенных возвратов
    "email": "<адрес электронной почты>"
    "phone": "<номер телефона>",
    "transactions": [ {
      "transactionId": "<номер транзакции>",
      "transactionStatusCode": "<код состояния транзакции>", -- см. секцию 4.2.17
      "transactionStatusText": "<название состояния транзакции>",
      "dateTime": "<дата проведения>",
      "cardNumber": "<маскированный номер карты>",
      "cardId": "<ИД привязанной карты>",
      "amount": "<сумма транзакции>"
    }, { ... } ] -- список оплаченных транзакций
  }
}
```

* при recurrent=true, если удалось создать шаблон платежа.

Рекуррентный платеж:

```
{
  "data": {
    "orderId": "<номер оплаты>",
    "amount": "<сумма оплаты>",
    "merchant": "<номер торговой точки>",
    "terminal": "<номер терминала>",
    "userId": "<ID пользователя мерчанта>",
    "orderStatusCode": "<код состояния оплаты>", -- см. секцию 4.2.10
    "orderStatusText": "<название состояния оплаты>"
    "createdRecurrentTemplateId": "<номер шаблона рекуррентного платежа>"
    "refunds": [], -- список проведенных возвратов
    "transactions": [
      {
        "dateTime": "<дата проведения>",
        "cardNumber": "<маскированный номер карты>",
        "cardId": "<ИД привязанной карты>",
        "amount": "<сумма оплаты>",
        "transactionId": "<номер транзакции>"
      },
      {
        "transactionStatusCode": "<код состояния транзакции>", -- см. секцию 4.2.17
        "transactionStatusText": "<название состояния транзакции>"
      }
    ],
    "email": "<адрес электронной почты>",
    "phone": "<номер телефона>"
  }
}
```

4.2.4 Запрос проведение рекуррентного платежа

Базовые параметры

URL	/recurrent
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
orderId	Номер заказа (не должен совпадать с номером заказа, по которому рекуррентный платёж был создан)	Уникальный для терминала, числовой, 1-50
amount	Сумма заказа в рублях и копейках. Может отличаться от суммы оригинального заказа.	Числовой, с двумя знаками после точки, > 0.00
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой, 1-50
recurrentTemplateId	Номер шаблона для создания рекуррентного платежа. Возвращается при успешном создании автоплатежа в ответе на запрос проведения оплаты (см. секцию 4.2.1)	Символьный, 1-255
recurrentInitiator	Инициатор автоплатежа. Допустимые значения: СІТ – инициатор клиент, МІТ_N – инициатор Торговая точка, Где значение N: 1 - плата за неявку или за неотмененное гарантированное бронирование; 2 - периодический платеж по поручению держателя карты без графика; 3 - периодический платеж по поручению держателя карты по графику.	Символьный, 1-255 Опциональный
sign	Подпись запроса	Символьный,

64 (для HMACSHA256)

Успешный ответ содержит JSON-структуру:

```
{
  "data": {
    "orderId": "<номер нового заказа>",
    "amount": "<сумма оплаты>",
  }
}
```

Не успешный ответ содержит JSON-структуру с кодом ответа:

```
{
  "data": {
    "code": "<ISO код ошибки>",
    "error": "<Текст ошибки>",
    "orderId": "<номер нового заказа>",
    "amount": "<сумма оплаты>",
  }
}
```

Коды ответа записываются в поле code.

Коды ответа соответствуют ISO 8583. Если эквайеры используют собственные коды ответов, то данные коды соотносятся с ISO 8583 по мере возможности.

4.2.5 Запрос на передачу данных для информирования ГИС ЖКХ

Данный запрос выполняется, если необходимо передавать информацию в ГИС ЖКХ о совершении платежа в пользу поставщиков ЖКХ услуг и в запросе на оплату флаг «gis» был передан со значением «true».

Необходимо отправить столько запросов, сколько услуг от разных поставщиков услуг ЖКХ, оплатил пользователь.

Базовые параметры

URL	/api/gis/
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
orderNumber	Номер оплаты.	Строковый, 1-50
amount	Сумма заказа в рублях и копейках. Может отличаться от суммы оригинального заказа.	Числовой, с двумя знаками после точки, > 0.00
terminalNumber	Номер терминала Торговой точки	Числовой, 1-50

account	Лицевой счёт абонента у поставщика ЖКХ услуг	Строковый, 1-30
bik	БИК банка получателя денежных средств за услугу ЖКХ	Числовой, 9
bankName	Наименование банка получателя денежных средств за услугу ЖКХ	Строковый, 1-170
bankAccount	Корсчёт банка получателя денежных средств за услугу ЖКХ	Числовой, 20
recipientAccount	Расчётный счёт получателя денежных средств за услугу ЖКХ	Числовой, 20
recipientInn	ИНН получателя денежных средств за услугу ЖКХ	Числовой, 10/12
recipientKpp	КПП банка получателя денежных средств за услугу ЖКХ. Значение «0», если получатель ИП	Числовой, 9
recipientKbk	КБК услуги	Числовой, 20 Оptionальный
recipientName	Наименование получателя денежных средств за услугу ЖКХ (юридическое лицо)	Строковый, 1-170
recipientOkato	ОКАТО получателя денежных средств за услугу ЖКХ	Числовой, 8 Оptionальный
yearOfPaymen	Год периода оплаты	Числовой, 4
monthOfPayment	Месяц периода оплаты	Числовой, 2
paymentPurpose	Назначение платежа. Например, «Оплата ЖКХ услуги»	Строковый, 1-210
sign	Подпись запроса	Символьный, 64 (для HMACSHA256)

4.2.6 Запрос на блокировку средств

Базовые параметры

URL	/blockpage
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
orderId	Номер заказа	Уникальный для терминала, числовой, 1-50
amount	Сумма заказа в рублях	Числовой, с двумя знаками после точки, > 0.00
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой, 1-50
clientBackUrl	URL для возврата на сайт Торговой точки	Символьный, 1-255
recurrent	Флаг необходимости проведения автоплатежа	Символьный (TRUE/FALSE) Оptionальный
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

4.2.7 Запрос на списание заблокированных средств**Базовые параметры**

URL	/charge
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
orderId	Номер заказа	Уникальный для терминала, числовой, 1-50
amount	Сумма заказа в рублях	Числовой, с двумя знаками после точки, > 0.00
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой,

		1-50
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

Успешный ответ содержит JSON-структуру:

```
{
  "data":{
    "amount":"<Сумма заказа>",
    "desc":"<Описание заказа>",
    "merchant":"<Номер мерчанта>",
    "orderId":"<Номер заказа>",
    "rc":"0",
    "terminal":"<Номер терминала>"
    "sign":"<Подпись ответа>"
  }
}
```

4.2.8 Запрос на разблокировку средств

Базовые параметры

URL	/retrieve
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
orderId	Номер заказа	Уникальный для терминала, числовой, 1-50
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой, 1-50
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

Успешный ответ содержит JSON-структуру:

```
{
  "data":{
    "amount":"<Сумма заказа через точку>",
    "desc":"<Описание заказа>",
    "merchant":"<Номер мерчанта>",
    "orderId":"<Номер заказа>",
    "rc":"0",
    "terminal":"<Номер терминала>"
    "sign":"<Подпись ответа>"
  }
}
```

```
}
}
```

4.2.9 Запрос на привязку карты

Базовые параметры

URL	/attachpage
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
orderId	Номер заказа	Уникальный для терминала, числовой, 1-50
amount	Сумма заказа в рублях	Числовой, с двумя знаками после точки, > 0.00 Необязательный. В случае отсутствия будет списана сумма по умолчанию 1, 00.
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой, 1-50
clientBackUrl	URL для возврата на сайт Торговой точки	Символьный, 1-255
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

Примечание: При успешном проведении операции сумма заказа будет заблокирована на карте клиента и сразу же разблокирована. Мерчант в HTTP-уведомлении об оплате (см. п. 2.3.1) получит значение номера шаблона для последующих рекуррентных операций (см. п. 4.2.4) с карты клиента.

4.2.10 Запрос на получение списка сохранённых карт

Базовые параметры

URL	/api/userid/cards
Тип запроса	GET

Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded
--------------------------	---

Параметры запроса

Название	Описание	Ограничения, формат, длина
userId	Уникальный идентификатор пользователя торговой точки.	Символьный, 1-50
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой, 1-50
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

Успешный ответ содержит JSON-структуру:

```
{
  [{"maskedPan": "<маскированный номер карты 1>", "cardId": "<ИД карты 1>", "paymentSystem": "<название МПС 1>"},
  {"maskedPan": "<маскированный номер карты 2>", "cardId": "<ИД карты 2>", "paymentSystem": "<название МПС 2>"},
  {"maskedPan": "<маскированный номер карты 3>", "cardId": "<ИД карты 3>", "paymentSystem": "<название МПС 3>"}],
}
```

4.2.11 Запрос для сохранения карты**Базовые параметры**

URL	/api/userid/card
Тип запроса	PUT
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
orderId	Номер заказа	Уникальный для терминала, числовой, 1-50
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой, 1-50
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

Успешный ответ содержит JSON-структуру:

```
{
  "orderId": "<номер оплаты>",
  "merchant": "<номер торговой точки>",
  "terminal": "<номер терминала>",
  "userid": "<ID пользователя мерчанта>",
  "maskedPan": "<Маскированный номер карты>",
  "cardId": "<ID карты>"
}
```

4.2.12 Запрос для удаления карты**Базовые параметры**

URL	/api/userid/card
Тип запроса	DELETE
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
userId	Уникальный идентификатор пользователя торговой точки.	Символьный, 1-50
cardId	Уникальный идентификатор ранее сохранённой карты пользователя торговой точки.	Числовой, 1-50 Опциональный
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой, 1-50
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

4.2.13 Запрос для оплаты дополнительными способами через API

Данный запрос используется для оплаты через кнопки быстрой оплаты, размещённые на странице Торговой точки.


Доступные способы оплаты (после согласования и включения их Мерчанту): GOOGLE PAY, SBERPAY, YANDEXPAY, SBP, MIRPAY.

Базовые параметры

URL	/api/token/payment/
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
orderId	Номер заказа	Уникальный для терминала, числовой, 1-50
amount	Сумма заказа в рублях и копейках	Числовой, с двумя знаками после точки, > 0.00
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой, 1-50
clientBackUrl	URL для возврата на сайт Торговой точки	Символьный, 1-255 Опциональный
recurrent	Флаг, определяющий необходимость создания рекуррентного платежа. По умолчанию создание отключено (соответствует отсутствию параметра или значению параметра false). При необходимости создания рекуррентного платежа в данном параметре необходимо передавать true	Логический, true/false Опциональный
userIp	IP-адрес клиента, передающего данные держателя карты	Символьный
tokenType	Передавать значение используемого способа: <ul style="list-style-type: none"> • GOOGLE_PAY, • SBER_PAY, • YANDEX_PAY • SBP • MIR_PAY 	Символьный
sendDeclinedNotification	Флаг, определяющий необходимость отправки уведомления об отклонённой транзакции при оплате способом SBER_PAY	Логический, true/false Опциональный
declinedNotificationURL	URL для отправки HTTP-уведомления об отклонённой	Символьный, 1-255 Опциональный

	транзакции при оплате способом SBER_PAY См. п. 2.3.2	
userOperationSystem	Операционная система устройства при оплате с моб устройства способом SBER_PAY. Допустимые значения: ANDROID I_OS	Символьный
mobileDeepLinkBackUrl	URL для возврата на сайт Торговой точки из приложения СБОЛ при оплате способом SBER_PAY. Передаётся в token .	Символьный
userOperationSystemType	Операционная система устройства при оплате с моб устройства способом SBER_PAY. Передаётся в token . Допустимые значения: ANDROID I_OS	Символьный
deviceOrigin	Тип устройства, с которого происходит оплата. Передаётся при оплате способами SBER_PAY и MIR_PAY. Передаётся в token . Допустимые значения: DESKTOP MOBILE MOBILE_APP	Символьный
token	Токен представляет собой JSON, закодированный в Base64. См. примечание и описание ниже. Пример:  token_JSON.txt	Символьный
phoneNumber	Номер телефона плательщика в СБОЛе. Передаётся в token при оплате SBER_PAY для получения push от СБОЛ.	Цифровой формат "^\(\\+?7 8)?([0-9]){10}\$" 10 символов Опциональный
sign	Подпись запроса	Символьный, 64 (для HMAC SHA256)

Примечание:

- 1) При оплате GOOGLE_PAY или YANDEX_PAY для параметра token все данные, кроме параметра userInfo, собираются с кнопки Google Pay/YandexPay. Данные для параметра userInfo мерчант собирает самостоятельно с браузера/устройства клиента.
- 2) При оплате SBER_PAY в параметре token передаётся объект, аналогичный Google Pay, но вместо токена от Google используется phoneNumber плательщика (может не передаваться, если не нужен push). Данные для параметра userInfo мерчант собирает самостоятельно с браузера/устройства клиента.
- 3) При оплате SBP в параметре token необходима передача параметра userInfo, данные для параметра userInfo мерчант собирает самостоятельно с браузера/устройства клиента.
- 4) При оплате MIR_PAY в параметре token необходима передача параметра userInfo, данные для параметра userInfo мерчант собирает самостоятельно с браузера/устройства клиента.

colorDepth	Глубина цвета в битах. Рекомендация по получению значения из браузера (из глобального объекта screen): screen.colorDepth	Символьный,
language	Язык браузера по формату IETF BCP47 Рекомендация по получению значения из браузера (из глобального объекта navigator): navigator.language	Символьный,
screenHeight	Высота экрана в пикселях. Рекомендация по получению значения из браузера (из глобального объекта screen): screen.height	Символьный,
screenWidth	Ширина экрана в пикселях. Рекомендация по получению значения из браузера (из глобального объекта screen): screen.width	Символьный,
timezone	Time-zone пользователя. Рекомендация по получению значения из браузера: вызов метода getTimezoneOffset()	Символьный, Пример: UTC +5 hours: -300
userAgent	Содержание заголовка User-Agent из браузера пользователя	Символьный,
browserAccept	Заголовок Accept из браузера пользователя	Символьный,

javaEnabled	Поддержка браузером пользователя Java	Символьный, (TRUE/FALSE) Опциональный
-------------	---------------------------------------	--

Успешный ответ содержит JSON-структуру:

Без 3ds:

```
{
  "paramsMap":{
    "amount":"<Сумма заказа через точку>",
    "desc":"<Описание заказа>",
    "createdRecurrentTemplateId ":"<Номер шаблона для создания рекуррентных платежей>"
    "merchant":"<Номер мерчанта>",
    "orderId":"<Номер заказа>",
    "originalTransactionId":"<Номер транзакции>",
    "rc":"0",
    "qrCodePaymentUrl":"<Ссылка для оплаты по СБП или SberPay>",
    "qrCodeOriginalPaymentUrl":"<Оригинальная ссылка НСПК для QR СБП>",
    "qrCodeContent":"<Изображение QR в кодировке base64>",
    "qrCodeMediaType":"<Формат изображения QR для оплаты СБП или SberPay>",
    "sign":"<Подпись ответа>",
    "terminal":"<Номер терминала>"
  }
}
```

С 3ds:

```
{
  "paramsMap":{
    "amount":"<Сумма заказа через точку>",
    "desc":"<Описание заказа>",
    "createdRecurrentTemplateId ":"<Номер шаблона для создания рекуррентных платежей>"
    "merchant":"<Номер мерчанта>",
    "orderId":"<Номер заказа>",
    "originalTransactionId":"<Номер транзакции>",
    "rc":"<502, 503, 504>",
    "threeDSMethodURL":"<URL для 3ds>"
    "sign":"<Подпись ответа>",
    "terminal":"<Номер терминала>"
  }
}
```

Примечание: Если операция происходит с 3ds, то в ответе будет код rc 502, 503 или 504. В параметре threeDSMethodURL передается URL, на который необходимо отправить пользователя для прохождения 3ds.

При оплате с мобильного устройства способом СБП необходимо отправить пользователя на URL, переданный в параметре qrCodePaymentUrl.

Не успешный ответ содержит JSON-структуру:

```
{
  "paramsMap":{
    "rc":"<Код ошибки>",
  }
}
```

4.2.14 Запрос доступности дополнительных способов оплаты через API

Базовые параметры

URL	/api/token/payment/activity/
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой, 1-50
tokenType	Тип токена/способа оплаты. Возможные значения: <ul style="list-style-type: none"> • GOOGLE_PAY • SBER_PAY • YANDEX_PAY • SBP • MIR_PAY 	Опциональный
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

Успешный ответ содержит JSON-структуру:

```
{
  "paramsMap":{
    "availableTokenTypes":"<Доступные способы оплаты токеном>",
    "googlePayType":"<Для типа оплаты токеном Google: PAN_ONLY, CRYPTOGRAM_3DS>",
    "merchant":"<Номер мерчанта>",
    "rc":"0",
    "sign":"<Подпись ответа>",
    "terminal":"<Номер терминала>"
  }
}
```

Не успешный ответ содержит JSON-структуру:

```
{
  "paramsMap":{
    "merchant":"<Номер мерчанта>",
    "rc":"242",
    "sign":"<Подпись ответа>",
    "terminal":"<Номер терминала>"
    "availableTokenTypes":"",
  }
}
```

Если оплата токеном недоступна, то в ответе будет код rc 242.

4.2.15 Запрос отмены оплаты (SberPay)

Базовые параметры

URL	/api/token/payment/decline/sber-pay/
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание	Ограничения, формат, длина
merchant	Номер Торговой точки	Числовой, 1-50
terminal	Номер терминала Торговой точки	Числовой, 1-50
orderId	Номер заказа	Уникальный для терминала, числовой, 1-50
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

Успешный ответ содержит JSON-структуру:

```
{
  "paramsMap": {
    "merchant": "777",
    "terminal": "1000",
    "orderId": "123",
    "rc": "244",
    "sign": "sign"
  }
}
```

Не успешный ответ содержит JSON-структуру:

```
{
  "paramsMap": {
    "merchant": "777",
    "terminal": "1000",
    "orderId": "123",
    "rc": "500",
    "sign": "sign"
  }
}
```

4.2.16 Коды и тексты состояний оплаты

orderStatusCode	orderStatusText	Описание
0	Создан	Пользователь перешёл на страницу оплаты
1	В обработке	Пользователь инициировал оплату
2	Оплачен	Оплата прошла успешно

4	Просрочен	Заказ не был оплачен за отведённое время
---	-----------	--

4.2.17 Коды и тексты состояний транзакций

transactionStatusCode	transactionStatusText	Описание
8	Оплачена	Транзакция оплачена
11	Возвращена	Транзакция возвращена
6	Блокирована	Средства заморожены
7	Списана	Средства списаны
10	Разблокирована	Средства разморожены
12	Просрочена	Транзакция не оплачена
9	Отменена	Транзакция отклонена

Возвращаемые ошибки

Код HTTP	Причина
400 Bad Request	Неверно указаны параметры запроса
401 Unauthorized	Запрос не аутентифицирован (неверная подпись HMAC)
404 Not Found	Заказ не найден по указанным параметрам

4.2.18 Коды ответов на запросы по API

Общие принципы

Код ответа записывается в поле гс.

Коды ответа до 200 соответствуют **ISO 8583**. Если эквайеры используют собственные коды ответов, то данные коды соотносятся с ISO 8583 по мере возможности.

В случае, если ответ эквайера нельзя соотнести с ISO 8583, код будет равен 501.

Коды ответов с 201 соответствуют ошибкам обработки запросов платежным шлюзом «ВсеПлатежи».

Код	Расшифровка
0	Успешное проведение операции
201	Сумма меньше либо равна нулю
202	Сумма имеет неверный формат
203	Ссылка для возврата к мерчанту не указана
204	Ссылка для возврата к мерчанту имеет неверный формат
205	Email имеет неверный формат

206	Описание платежа имеет неверный формат
207	Идентификатор плательщика имеет неверный формат
208	Номер мерчанта или номер терминала имеет неверный формат
209	Номер платежа не указан
210	Номер платежа имеет неверный формат
211	Данный тип интеграции не поддерживается
212	Идентификатор сессии плательщика не указан
213	Терминал мерчанта или мерчант не найден
214	Платёж с таким номером уже существует
215	Платёж с таким номером не найден
216	Терминал мерчанта отключен
217	Средства не были заблокированы
218	В настоящее время уже выполняется списание средств
219	По данному платежу уже было выполнено списание средств
220	В настоящее время уже выполняется разблокировка средств
221	В настоящее время уже выполняется процесс оплаты
222	В настоящее время уже выполняется блокировка средств
223	Сумма не соответствует ожидаемой
224	Неверный номер карты
225	Карта просрочена
226	MD имеет неверный формат
227	Указан неверный MD
228	Результат с 3DS не ожидается
229	Операция не ожидается
230	Неверные данные карты
231	IP адрес клиента указан не верно
232	Невалидная подпись
233	Не найден шаблон для автоплатежа
234	Номер телефона имеет неверный формат
235	Возврат для данного терминала запрещён
236	Один из дополнительных параметров имеет неверный формат
237	Один из дополнительных параметров не ожидается
238	Один из обязательных дополнительных параметров не был передан
239	Заказ просрочен
240	Не пройдена проверка 3ds

241	Переданный токен имеет не корректный формат
242	Оплата данным видом токена временно не доступна
243	Сумма оплаты превышает допустимый лимит
244	Транзакция SberPay успешно отклонена
245	Произошла непредвиденная ошибка при получении ссылки для оплаты. Попробуйте повторить запрос позднее
500	Внутренняя ошибка
501	Ошибка на стороне эквайера
502	Необходимо пройти 3ds
503	Необходимо позволить ACS собрать данные пользователя
504	Необходимо пройти 3ds второй версии

4.3 Алгоритм формирования НМАС

НМАС должен проверяться Платёжным шлюзом при получении запроса на проведение операции от Торговой точки и при получении ответа по результату проведения операции от Платёжного шлюза Торговой точкой. НМАС формируется в два этапа: 1 – формирование строки данных для НМАС, 2 – формирование НМАС.

4.3.1 Подготовка строки данных для НМАС

Этапы:

1. Каждое значение параметра дополняется его длиной: «длина текстового значения параметра в байтах» + «значение» (например, значение 1000.00 для использования в подписи будет иметь вид 71000.00, <https://vp.ru> → 13<https://vp.ru>, а оплата услуги → 25оплата услуги, так как русские буквы в кодировке UTF-8 занимают по два байта, плюс пробел – один байт).
2. Имена параметров должны быть отсортированы в алфавитном порядке.
3. Значения параметров, полученные в п.1, соединяются в одну строку без разделителей в порядке следования отсортированных именованных.

Важно:

- Не нужно выполнять кодирование URL (encoding) или экранирование HTML-символов (escaping). Например, знак & не должен преобразовываться в &.
- Для значений параметров должна использоваться кодировка UTF-8.
- Параметр `sign` на этапе подготовки строки для подписи не используется.
- Если значение параметра в байтах равно 0, то параметр в строке для подписи не используется.

Пример:

Допустим, Пользователь инициировал процесс оплаты, и Торговая точка имеет следующие значения для запроса с целью проведения операции оплаты:

```
orderId=10000000001
amount=100.00
merchant=777
terminal=1001
clientBackUrl=https://example-merchant:8081/back-from-pay
description=Оплата за электроэнергию
userid=101
```

По условиям, чтобы подготовить значения параметров для подписи, необходимо выстроить их по названию параметров:

```
amount=100.00
clientBackUrl=https://example-merchant:8081/back-from-pay
description=Оплата за электроэнергию
merchant=777
orderId=10000000001
terminal=1001
userid=101
```

Затем значения параметров преобразуются и складываются:

```
6100.0043https://example-merchant:8081/back-from-pay46Оплата за электроэнергию3777110000000001410013101
```

Строка готова для генерации подписи.

4.3.2 Генерация HMAC

Для формирования HMAC используется алгоритм «HMACSHA256». В качестве ключа используется секретный ключ терминала Торговой точки. Ключ представляет собой последовательность байт в HEX-формате, например:

```
b22ec899aaf398624c14305d56a3aa98095523fe
```

Но для формирования HMAC он должен быть преобразован в бинарный массив. Данный ключ для наглядности можно представить в десятичном формате таким образом:

```
[178, 46, 200, 153, 170, 243, 152, 98, 76, 20, 48, 93, 86, 163, 170, 152, 9, 85, 35, 255]
```

Подпись HMAC для строки из предыдущего примера с использованием данного ключа имеет вид:

```
5d3973c71f2fc12e8b1ff91dad63b58c7e377cccbcd6bf01d3621ab3bd44189d.
```

4.4 Примеры для некоторых языков программирования

4.4.1 PHP

```
$stringToSign = '510.0144https://example-merchant:8081/pay-
  result=20046Оплата за электроэнергию37771110000000001410013101';
$shared_key = 'b22ec899aaf398624c14305d56a3aa98095523ff';
$shmac = hash_hmac('SHA256', $stringToSign, pack('H*', $shared_key));
```

Результат в \$shmac:

```
79c1947a8a9fced811af0a2f357aebdf027256761b926866eac65b4652323bcb
```

4.4.2 Java

```
import org.apache.commons.codec.binary.Hex;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
...
String stringToSign = '510.0144https://example-merchant:8081/pay-
  result=20046Оплата за электроэнергию37771110000000001410013101';
Mac mac = Mac.getInstance("HmacSHA256");
byte[] signingKeyBytes = Hex.decodeHex(sharedSecretKey.toCharArray());
SecretKeySpec signingKey =
  new SecretKeySpec(signingKeyBytes, mac.getAlgorithm());
mac.init(signingKey);
byte[] rawHmac = mac.doFinal(stringToSign.getBytes(StandardCharsets.UTF_8));
byte[] hexBytes = new Hex().encode(rawHmac);
String stringHmac = new String(hexBytes, StandardCharsets.UTF_8);
return stringHmac;
```

4.5 Дополнительные параметры в запросы по API

В зависимости от настроек системы, а также типе принимаемых платежей, схемы работы Торговой точки и требований банков эквайеров от мерчанта требуется передача дополнительных данных в запросах при проведении оплаты.

Параметры для запроса

Название	Описание	Ограничения, формат, длина
description	Описание заказа	Символьный, 1-255 Опциональный
notificationURL	URL для отправки HTTP-уведомления о совершённой оплате. См. п. 2.3.1	Символьный, 1-255 Опциональный

dark_mode	Флаг, определяющий необходимость отображения платёжной страницы в тёмном фоне. По умолчанию страница в светлом фоне (соответствует отсутствию параметра или значению параметра false). При необходимости страницы в тёмном фоне в данном параметре необходимо передавать true	Логический, true/false Оptionальный
gis	Флаг, определяющий необходимость информирования ГИС ЖКХ об оплате услуг ЖКХ. По умолчанию уведомление отключено (соответствует отсутствию параметра или значению параметра false). При необходимости уведомления ГИС ЖКХ в данном параметре необходимо передавать true	Логический, true/false Оptionальный
storename	Наименование магазина Торговой точки.	Символьный, 1-255 Оptionальный
savecard	Флаг, определяющий необходимость активации признака «Сохранить карту» на странице Платёжного шлюза.	Логический, true/false Оptionальный
userid	Уникальный идентификатор пользователя торговой точки. Может быть пустым, например, при проведении платежа без регистрации пользователя на сайте Торговой точки	Символьный, 1-50 Оptionальный
cardId	Уникальный идентификатор ранее сохранённой карты пользователя торговой точки. Передаётся в связке с userid для выбора по умолчанию сохранённой карты на странице Платёжного шлюза. Может быть пустым, например, при проведении платежа без регистрации пользователя на сайте Торговой точки	Числовой, 1-50 Оptionальный
email	Адрес электронной почты для квитанции или фискального чека. Если для терминала настроено отображение поля «Email», то	Символьный, формат: [a-zA-Z0-9+_.-]+@[a-zA-Z0-9-.-]+ 1-255 Оptionальный

	пользователю будет доступно изменение переданного значения.	
phone	Номер телефона плательщика отправки для фискального чека. Если для терминала настроено отображение поля «phone», то пользователю будет доступно изменение переданного значения.	Цифровой формат [0-9]{10} 10 символов Опциональный
operator_name	Наименование моб. оператора. Передаётся обязательно при оплате мобильной связи.	Символьный, 1-255 Опциональный
billing_phone	Номер телефона. Передаётся обязательно при оплате мобильной связи.	Цифровой формат [0-9]{10} 10 символов Опциональный
sendDeclinedTransactionNotification	Флаг, определяющий необходимость отправки HTTP-уведомления по неуспешным транзакциям.	Логический, true/false Опциональный
declinedTransactionNotificationUrl	URL для отправки HTTP-уведомления по неуспешным транзакциям.	Символьный, 1-255 Опциональный
user_commission	Сумма дополнительной комиссии с плательщика, взятая Торговой точкой.	Числовой, с двумя знаками после точки, > 0.00 Опциональный
merchantOrderId	Номер заказа на стороне Торговой точки	Символьный, 1-255 Опциональный
bic	Номер БИК банка. Передается обязательно по платежам в пользу погашения кредита.	Числовой, 9 Опциональный
checkingAccount	Номер расчётного счёта клиента. Передается обязательно по платежам в пользу погашения кредита.	Числовой, 20 Опциональный
inn	Значение ИНН получателя/поставщика услуг. Передается обязательно в случае приёма платежей Торговой точкой в пользу разных поставщиков услуг.	Числовой, 10,12 Опциональный
recipientName	Наименование получателя/поставщика услуг. Передается обязательно в случае	Символьный, 1-50 Опциональный

	приёма платежей Торговой точкой в пользу разных поставщиков услуг.	
recipientCity	Город получателя/поставщика услуг. Передается обязательно в случае приёма платежей Торговой точкой в пользу разных поставщиков услуг.	Символьный, 1-32 Опциональный

Раздел 5 Платёжный виджет

5.1 Общее описание

Платёжный виджет – самый простой способ подключения к Платёжному шлюзу для приёма платежей. Для этого на своём сайте Торговой точке необходимо вставить HTML и JavaScript-код, описание и пример для которых смотрите далее. Пример платёжной формы, открытой в виджете, показан на Рисунке 4.

The screenshot shows a payment form titled "Ввод данных для оплаты" (Enter payment data) with a 03:09 timer. The payment number is 945311492493333. The form includes fields for card number (0000 0000 0000 0000), expiration date (Month: 00, Year: 00), and a three-digit security code (000). There are checkboxes for "Получить квитанцию" (Get receipt) and a blue "Оплатить 11.01 Р" button (Pay 11.01 R) with a note "в том числе комиссия 1.00 Р" (including 1.00 R commission). A link "Отменить и вернуться" (Cancel and return) is also present. At the bottom, a disclaimer states: "Нажимая на кнопку «Оплатить», вы соглашаетесь с условиями [оферты](#)" (By clicking the "Pay" button, you agree to the terms of the [offer](#)).

Рисунок 4: Платёжный виджет, открытый на сайте Торговой точки

5.2 Подключение виджета

Далее представлен пример HTML-страницы с блоками кода, которые необходимы для работы

виджета. Эти блоки помечены комментарием `<!--[VsePlatezhi-Widget]-->`. Для использования кода на своём сайте необходимо:

1. Заменить значение `<адрес-сервера-шлюза>` на один из адресов, приведённых в п. 4.1.
2. Задать рабочие номера мерчанта и терминала.
3. Стилизовать форму (`name="VpGatewayForm"`) по своим требованиям. Данная форма непосредственно не относится к виджету, но она имеет необходимые для вызова виджета поля ввода (с определёнными атрибутами `name`), и настроена на вызов функции `makePayment()`, которая иницирует открытие виджета. В данном примере скрыты поля «номер заказа» и «описание», так как они необязательные, а отображается только поле «сумма» и кнопка открытия виджета.

Пример

```

<!DOCTYPE html>
<html lang="ru">

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no">

<!--[VsePlatezhi-Widget]-->
<script src="https://<адрес-сервера-шлюза>/resources/js/widget-script.js"></script>

<!--[VsePlatezhi-Widget]-->
<script>
function makePayment(amount, description, email, orderId) {
  var params = {
    merchant: '100001', // Номер мерчанта (пример)
    terminal: '10000100001', // Номер терминала (пример)
    amount: amount, // Сумма заказа, в формате 0.00 (форматируется автоматически)
    description: description, // Описание заказа (не обязательно)
    email: email, // Адрес электронной почты для получения квитанции (не обязательно)
    orderId: orderId, // Номер заказа (не обязательно)
    frame: true // Способ отображения платёжной формы:
      // true - во всплывающем блоке (фрейме),
      // false - в отдельном окне.
  };
  goPay(params);
}
</script>
</head>

<body>
<div>
<!--[VsePlatezhi-Widget]-->
<form class="vp-gateway-form" name="VpGatewayForm">
  <input type="hidden" class="vp-gateway-input orderId" name="orderId"
    placeholder="Номер заказа" />
  <input type="hidden" class="vp-gateway-input description" name="description"
    placeholder="Описание" />
  <input type="hidden" class="vp-gateway-input email" name="email" placeholder="Email" />
  <input type="tel" class="vp-gateway-input amount" name="amount"
    placeholder="Сумма" />
  <input type="button" class="vp-gateway-btn pay" value="Оплатить"
    onclick="makePayment(
      document.VpGatewayForm.amount.value,
      document.VpGatewayForm.description.value,
      document.VpGatewayForm.email.value,
      document.VpGatewayForm.orderId.value)" />

```

```
</form>  
  
</div>  
</body>  
  
</html>
```

5.3 Параметры вызова

Набор параметров для вызова страницы виджета базируется на параметрах API, см. п. 4.2.1, с некоторыми упрощениями:

1. Параметр `orderId` является опциональным. Если он не передан, то номер заказа будет сгенерирован автоматически.
2. Параметр `clientBackUrl` также является опциональным. Если он не передан, то пользователь по итогам оплаты будет перенаправлен на специальную страницу на сайте Платёжного шлюза, на которой будет отображена квитанция или сообщение, что оплата заказа была отменена.
3. Параметр `userid` не используется. Привязка карт не доступна пользователю.
4. Параметр `sign` не используется. Подписывать запрос не нужно.

Раздел 6 Реестр проведенных операций

6.1 Формирование и отправка реестра

Реестр проведенных операций формируется на следующие сутки и отправляется вложением в отдельном письме на адрес электронной почты, предоставленный Торговой точкой для этих целей.

6.2 Формат письма

Тема:

Реестр операций за ДД.ММ.ГГГГ г. <наименование торговой точки>

Содержание:

Реестр операций за ДД.ММ.ГГГГ г. <наименование торговой точки>

Количество записей в реестре: <фактическое количество записей в реестре>

Вложение:

Файл реестра.

6.3 Формат файла реестра

Реестр формируется в виде **.xlsx**-файла (в формате Excel 2007 и выше).

Наименование файла:

MERCH_ID_TERM_ID_ДД.ММ.ГГГГ.xlsx, где

- MERCH_ID – номер Мерчанта (Торговой точки);
- TERM_ID – номер терминала;
- ДД.ММ.ГГГГ – дата, за которую сформирован реестр.

Структура файла:


Наименование поля	Описание
Номер заказа	Номер заказа. Формат: текстовый.
Номер транзакции	Номер транзакции. Формат: текстовый.

Код авторизации	Код авторизации операции. Формат: текстовый.
Дата и время операции	Дата и время перехода транзакции в статус «Оплачена» в случае оплаты услуг или «Возвращена» в случае возврата денежных средств. Формат: дата и время (ДД.ММ.ГГГГ чч:мм:сс).
Номер карты	Маскированный номер карты. Формат: текстовый.
Тип операции	Доступные значения: <ul style="list-style-type: none"> • «Оплата» • «Возврат» Формат: текстовый.
Сумма операции	Общая сумма заказа или возврата. Значение всегда положительное, даже в случае возврата. Формат: числовой #0.00*.
Сумма к перечислению	Общая сумма операции за вычетом суммы комиссии за проведение операции. Сумма возврата включается в отчёт со знаком минус. Формат: числовой #0.00*.
Комиссия	Комиссия шлюза за проведение операции. Формат: числовой #0.00*.

Примечание:

* В зависимости от настроек операционной системы, суммы могут отображаться с точкой или с запятой в качестве десятичного разделителя. Для российской локали используется запятая, как показано в примере ниже.

Пример:

 1000_2000_09.08.2017.xlsx

	A	B	C	D	E	F	G	H	I
1	Реестр операций								
2	Номер заказа	Номер транзакции	Код авторизации	Дата и время операции	Номер карты	Тип операции	Сумма операции	Сумма к перечислению	Комиссия
3	112233000101	963019039	123456	09.08.2017 11:47:38	123456*****1234	Оплата	30,00	27,00	3,00
4	112233000102	963019456	123457	09.08.2017 12:06:02	123456*****1234	Оплата	1 000,00	970,00	30,00
5	112233000101	963019963	123456	09.08.2017 16:36:19	123456*****1234	Возврат	30,00	-30,00	0,00
6	Итого:						1 060,00	967,00	33,00

ВсеПлатежи Заказы Платежные ссылки Выйти

Создание платежной ссылки

Сумма заказа*
1200

Терминал*
▼

Номер заказа
1639557416864

Описание

Перенаправлять пользователя после оплаты

BackUrl
<https://mycompany.ru>

[Получить ссылку](#) [Сбросить](#)

Платежная ссылка
<https://go.vp.ru/77YP55> [Скопировать](#)

Отправить ссылку на e-mail Отправить ссылку в sms

Отправить ссылку на e-mail
e-mail [Отправить](#)

Рисунок 6: Страница создания ссылки