

Платёжный шлюз «ВсеПлатежи»

Руководство Мерчанта

Оглавление

| | |
|---|----|
| История изменений..... | 3 |
| Раздел 1 Определения | 5 |
| Раздел 2 Общие сведения | 7 |
| 2.1 Защищённое соединение | 7 |
| 2.2 Способы проведения оплаты | 8 |
| 2.3 Алгоритм проведения оплаты..... | 8 |
| 2.3.1 Уведомление об успешном проведении оплаты | 13 |
| Раздел 3 Платёжная форма | 14 |
| 3.1 Общий вид | 14 |
| 3.2 Основные элементы формы | 15 |
| Раздел 4 API | 17 |
| 4.1 Адреса серверов | 17 |
| 4.2 Поддерживаемые запросы..... | 17 |
| 4.2.1 Запрос на проведение оплаты | 18 |
| 4.2.2 Запрос статуса заказа | 20 |
| 4.2.3 Запрос расширенного статуса заказа..... | 21 |
| 4.2.4 Запрос проведение рекуррентного платежа | 23 |
| 4.2.5 Запрос на передачу данных для информирования ГИС ЖКХ..... | 24 |
| 4.3 Алгоритм формирования НМАС..... | 26 |
| 4.3.1 Подготовка строки данных для НМАС..... | 26 |
| 4.3.2 Генерация НМАС | 27 |
| 4.4 Примеры для некоторых языков программирования | 28 |
| 4.4.1 PHP..... | 28 |
| 4.4.2 Java..... | 28 |
| Раздел 5 Платёжный виджет | 29 |
| 5.1 Общее описание | 29 |
| 5.2 Подключение виджета..... | 30 |
| 5.3 Параметры вызова..... | 31 |
| Раздел 6 Реестр проведённых операций..... | 32 |
| 6.1 Формирование и отправка реестра..... | 32 |
| 6.2 Формат письма | 32 |
| 6.3 Формат файла реестра | 32 |

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 |
|--|--|---|

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.

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.

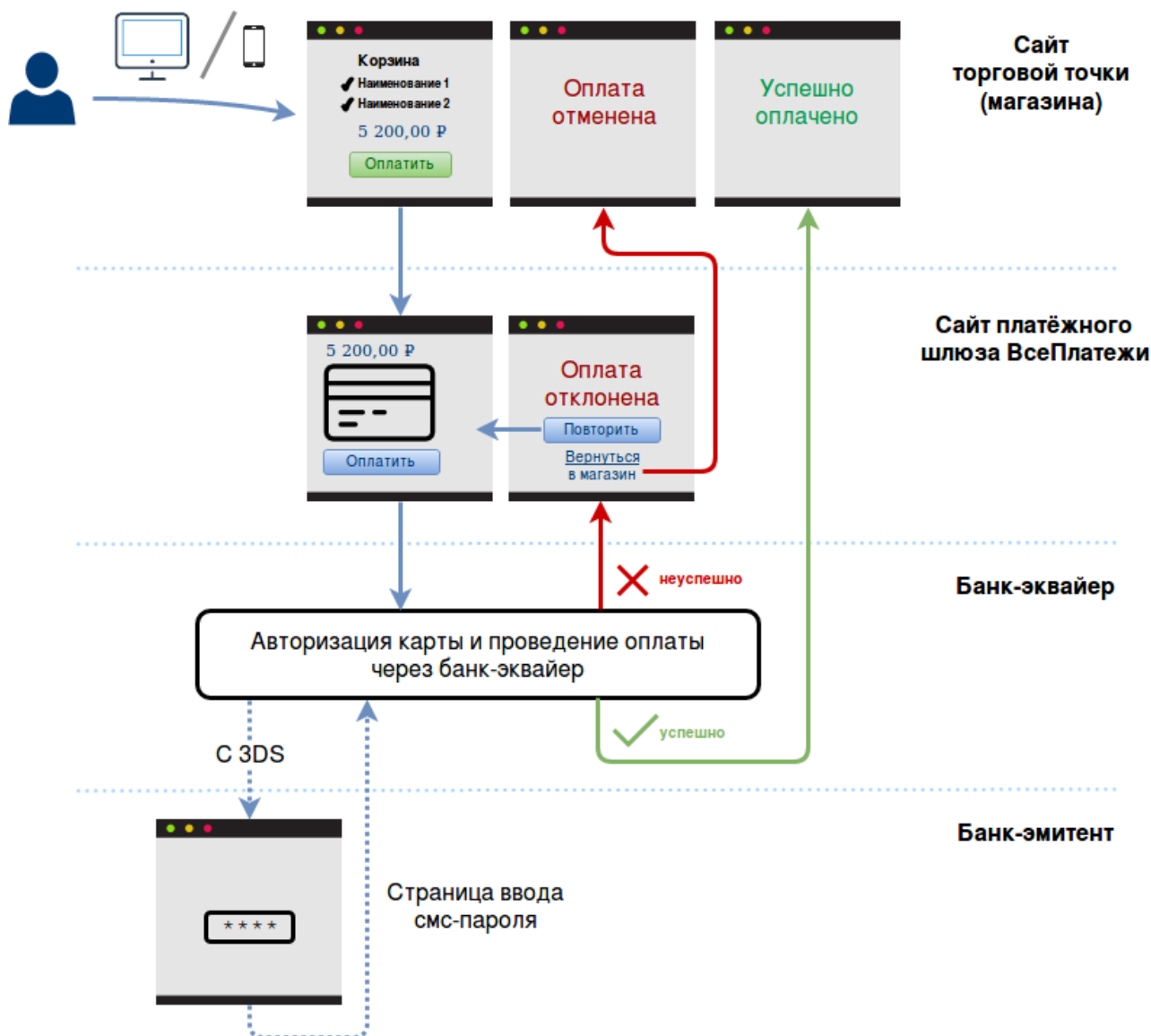


Рисунок **Ошибка! Только основной документ.**: Общая схема приёма платежа через API

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.

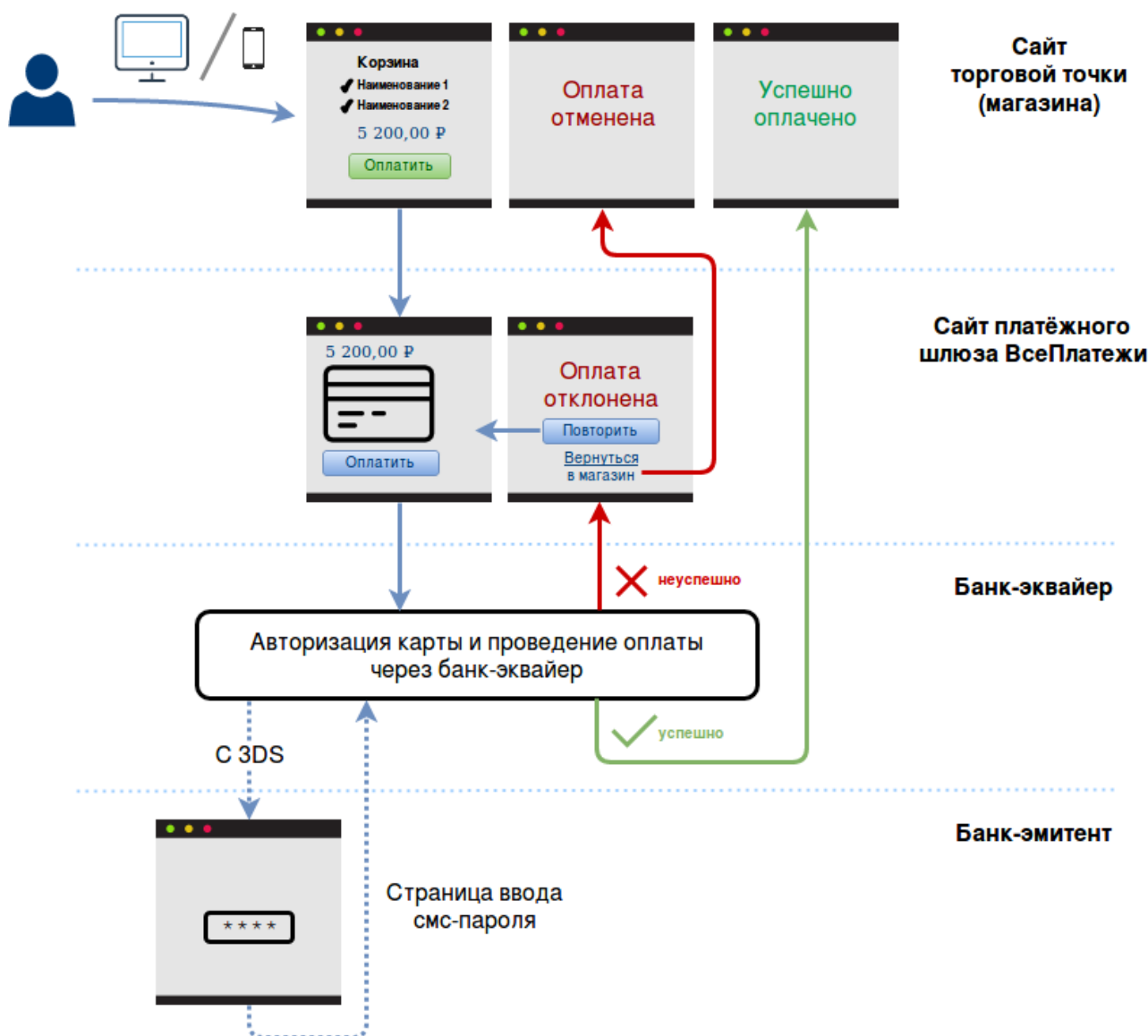


Рисунок **Ошибка! Только основной документ.**: Общая схема приёма платежа через API

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.

Подробнее смотрите в Раздел 4 и Раздел 5.

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

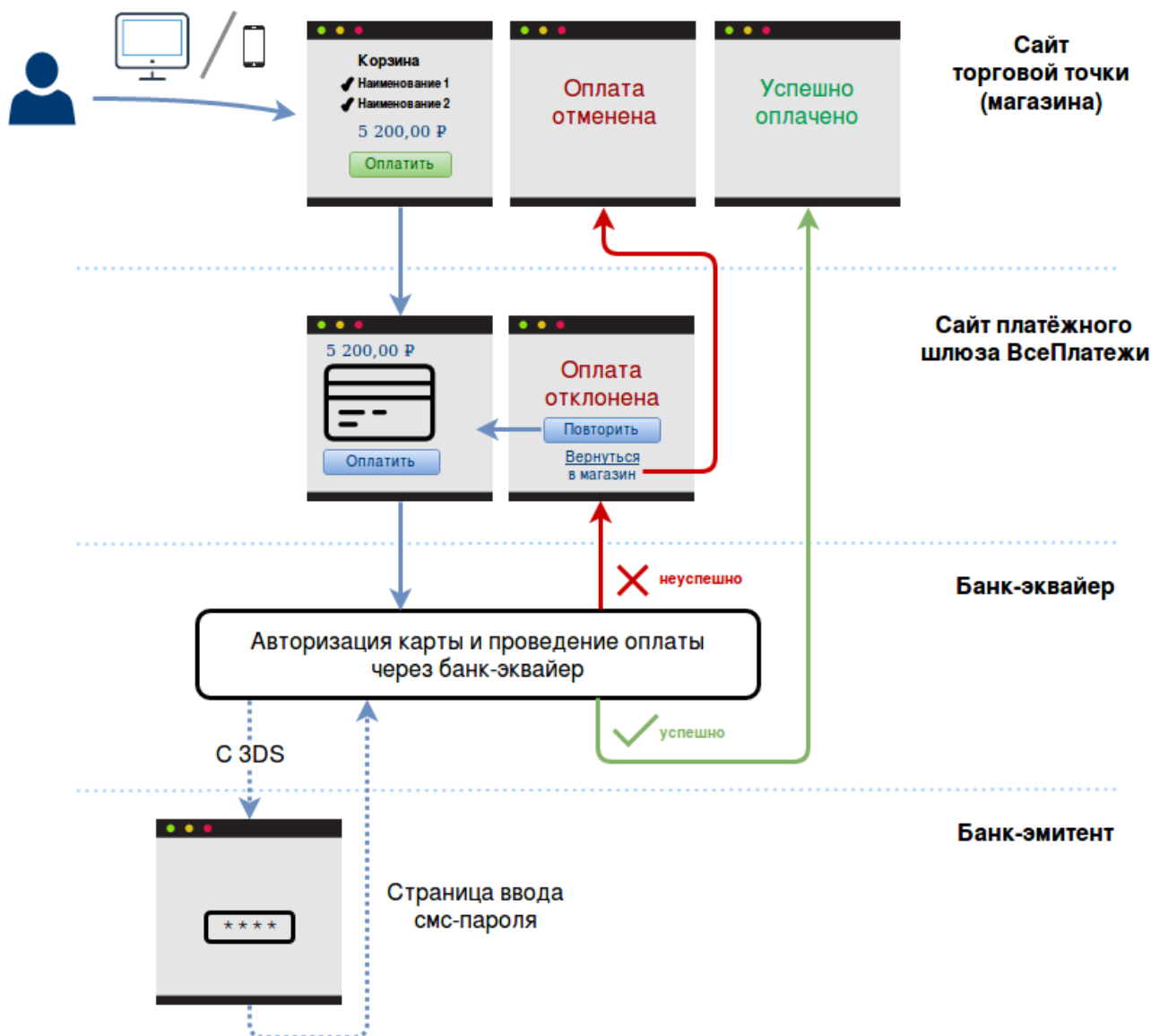


Рисунок **Ошибка! Только основной документ.**: Общая схема приёма платежа через API

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 `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).

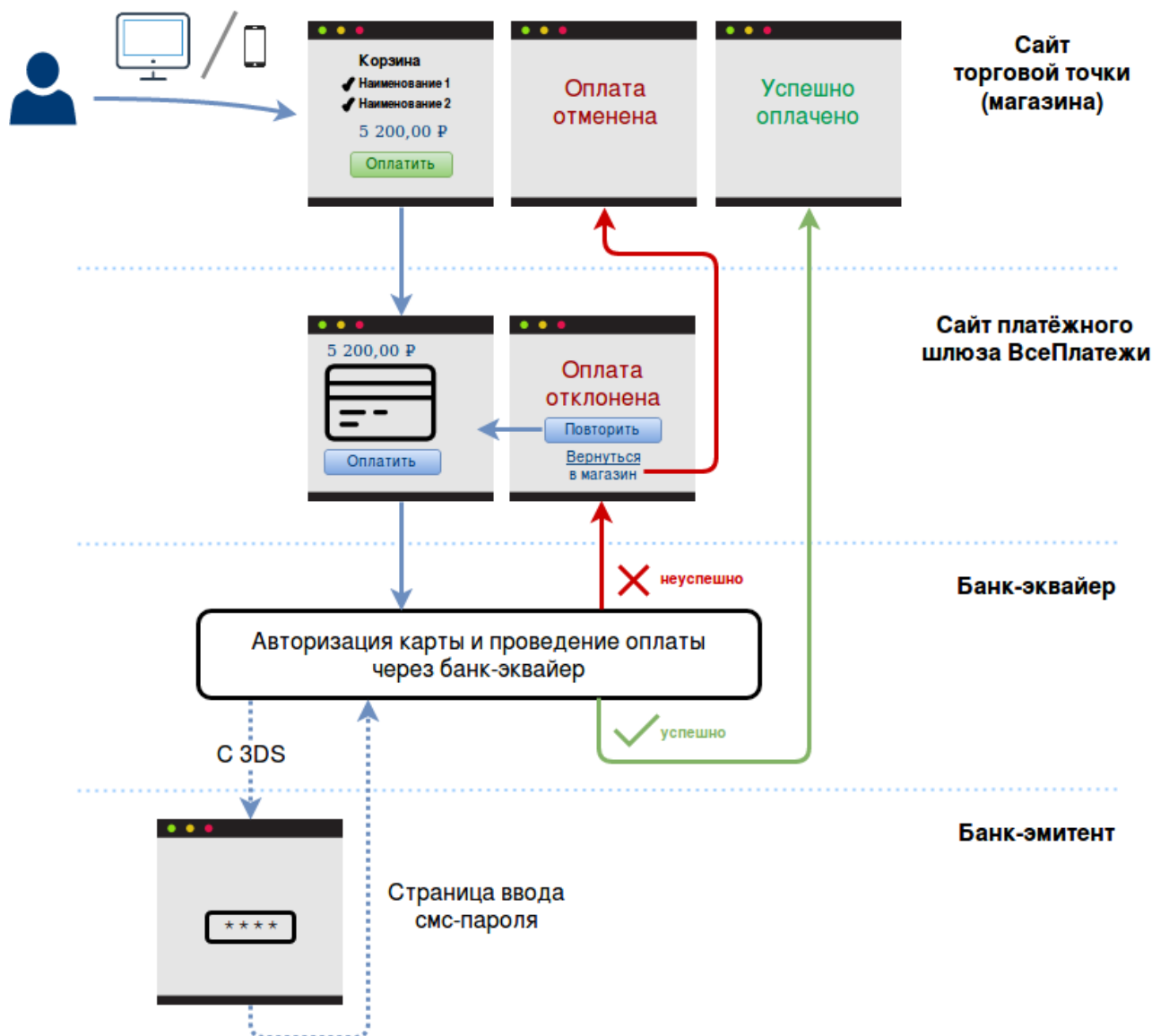


Рисунок **Ошибка! Только основной документ.**: Общая схема приёма платежа через API

3. 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.

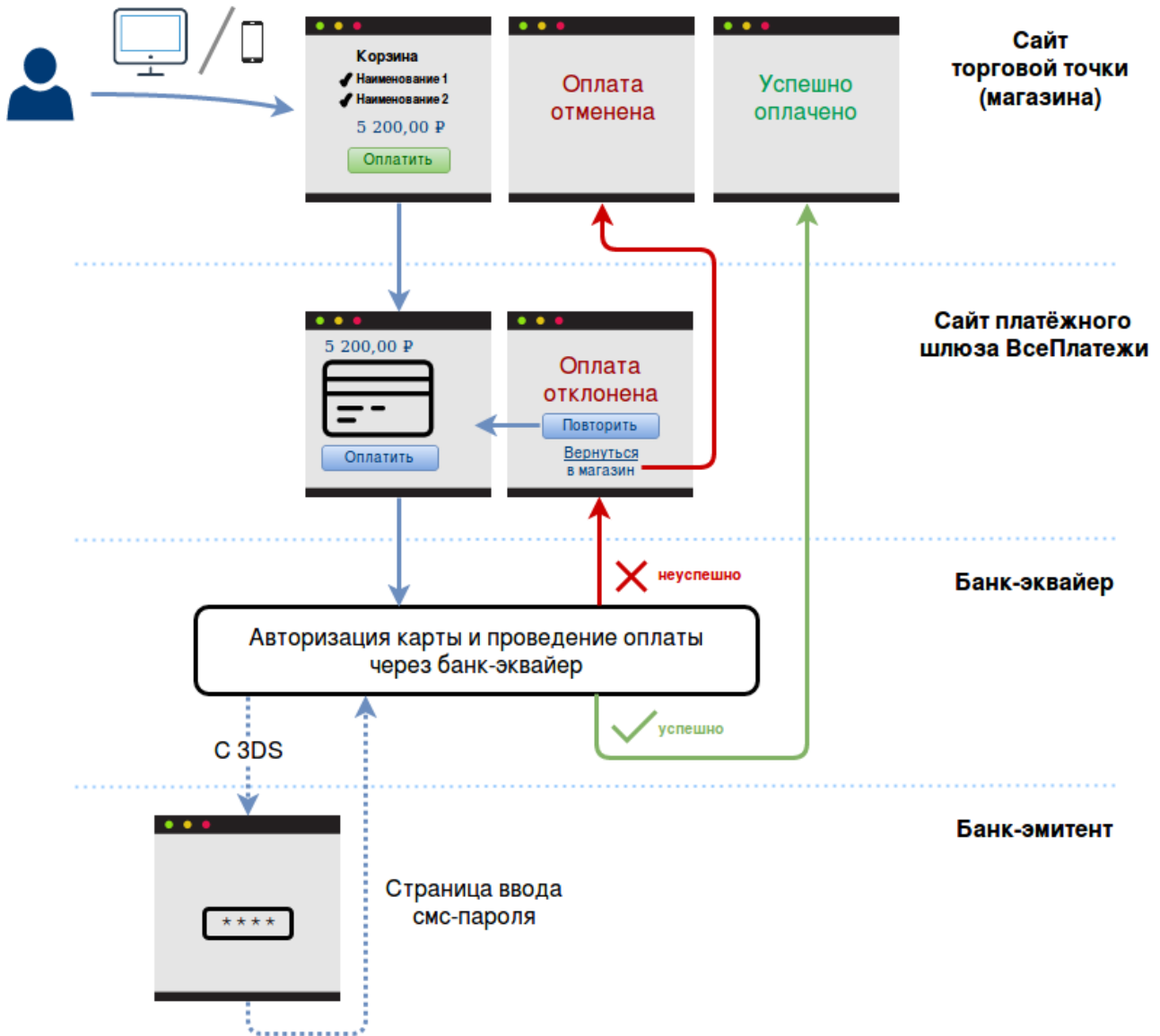
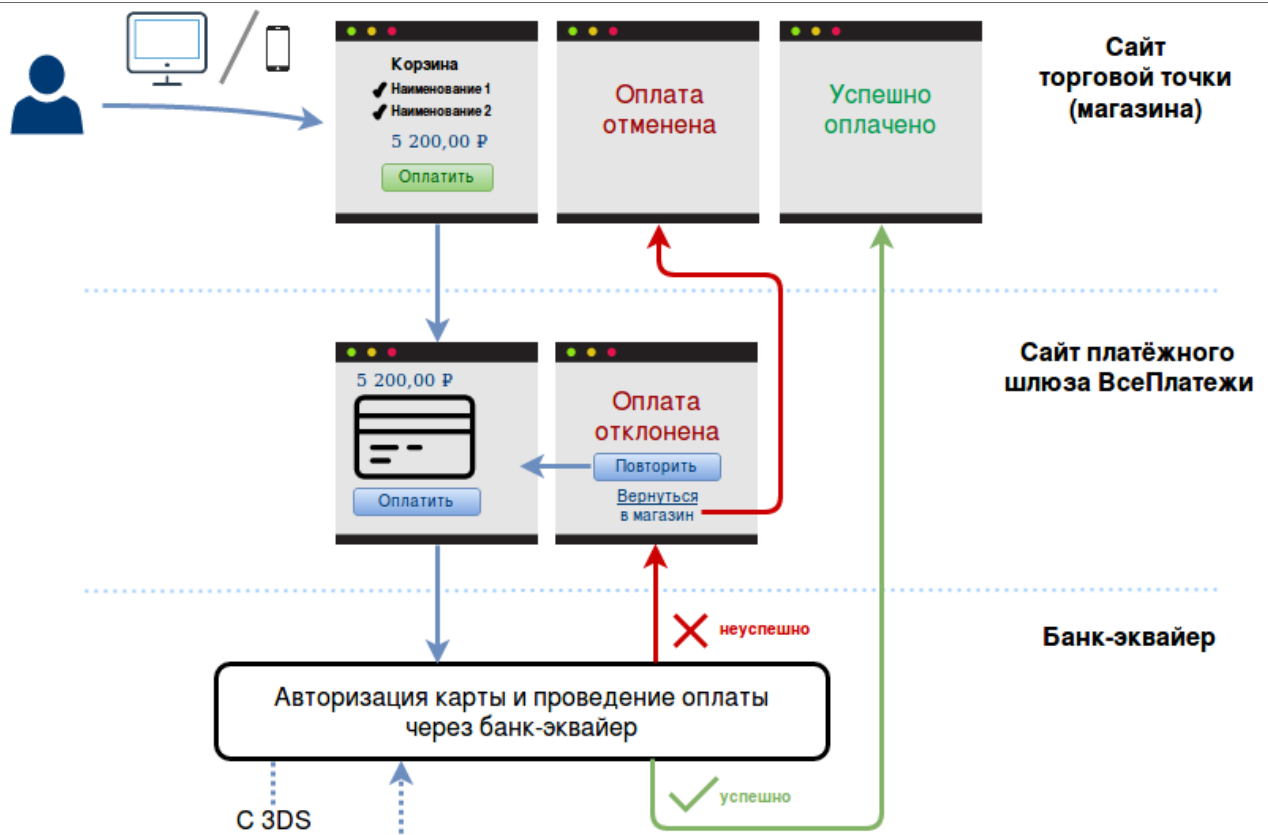


Рисунок **Ошибка! Только основной документ.:** Общая схема приёма платежа через API



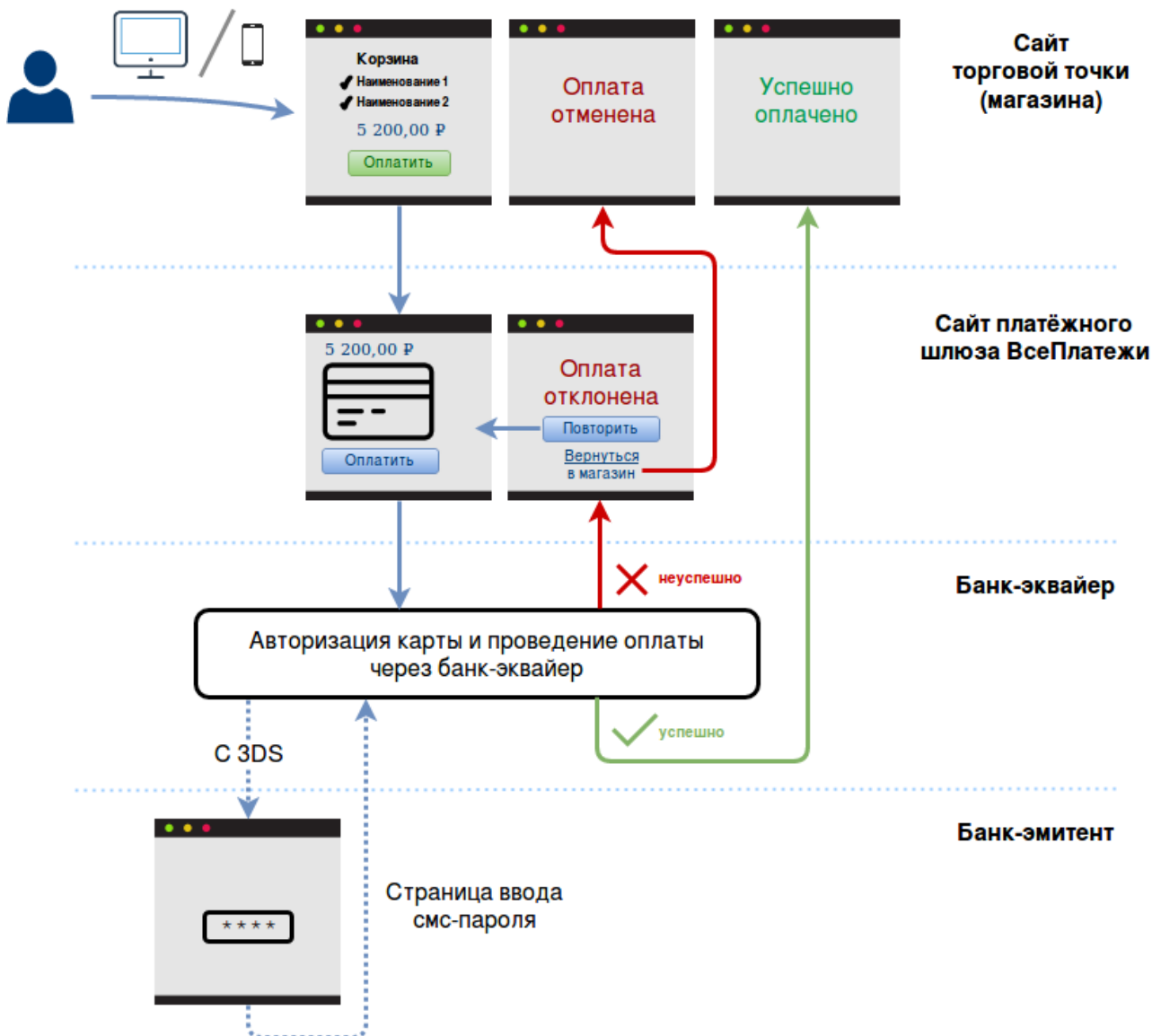


Рисунок **Ошибка! Только основной документ.:** Общая схема приёма платежа через API

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.

Список отправляемых параметров:

1. orderId
2. amount
3. terminal
4. merchant
5. transactionId
6. transactionDateTime
7. cardNumber
8. createdRecurrentTemplateId
9. email
10. phone
11. sign

Кроме параметра sign («подпись»), все остальные параметры берутся из инициирующего запроса на оплату, сформированного Торговой точкой. Подпись генерируется для указанных параметров по алгоритму, описанному в п. 4.3, 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».

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.

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. A card number field contains '0000 0000 0000 0000'. Below the number is a field for 'Срок действия карты' (card validity) with 'Месяц' (Month) and 'Год' (Year) sub-fields, both containing '00'. To the right of the validity field is a field for 'Три цифры на обороте' (Three digits on the back) 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

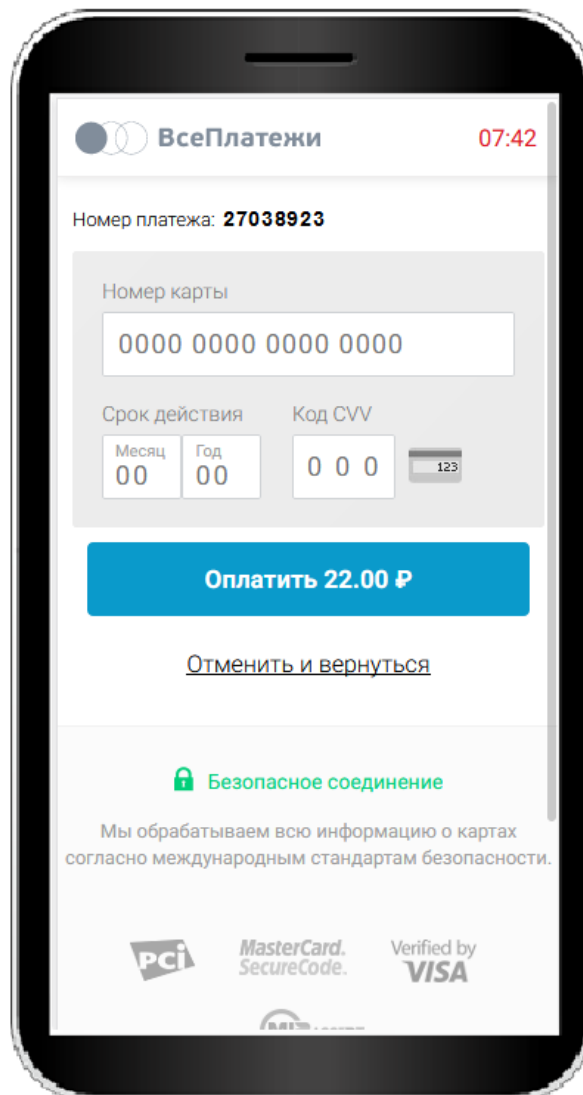


Рисунок 3: Вид платёжной формы на экране смартфона

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

| 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 рекуррентного платежа через API | POST /recurrent | <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 статуса заказа | POST /api/order/status | <i>Успешно:</i> |

| | | |
|--|----------------------------|---|
| | | <p>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, соответствующим 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, соответствующим типу ошибки.</p> |

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 |
| | Номер Торговой точки | Числовой, |

| | | |
|---------------|--|---|
| merchant | | 1-50 |
| terminal | Номер терминала Торговой точки | Числовой, 1-50 |
| clientBackUrl | URL для возврата на сайт Торговой точки | Символьный, 1-255 |
| description | Описание заказа | Символьный, 1-255 Опциональный |
| email | Адрес электронной почты. Если для терминала настроено отображение поля «Email», то пользователю будет доступно изменение переданного значения. | Символьный, формат: [a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+ 1-255 Опциональный |
| phone | Номер телефона. Если для терминала настроено отображение поля «phone», то пользователю будет доступно изменение переданного значения. | Символьный формат [0-9]{10} 10 символов Опциональный |
| userid | Уникальный идентификатор пользователя торговой точки. Может быть пустым, например, при проведении платежа без регистрации пользователя на сайте Торговой точки | Символьный, 1-50 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 |
| 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 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": {
    "orderNumber": "<номер оплаты>",
    "amount": "<сумма оплаты>",
    "merchantNumber": "<номер торговой точки>",
    "terminalNumber": "<номер терминала>",
    "userIdNumber": "<ID пользователя мерчанта>",
    "orderStatusCode": "<код состояния оплаты>", -- см. секцию 4.2.6
    "orderStatusText": "<название состояния оплаты>" -- см. секцию 4.2.6
    "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": {
    "orderNumber": "<номер оплаты>",
    "amount": "<сумма оплаты>",
    "merchantNumber": "<номер торговой точки>",
    "terminalNumber": "<номер терминала>",
    "userIdNumber": "<ID пользователя мерчанта>",
    "orderStatusCode": "<код состояния оплаты>", -- см. секцию 4.2.6
    "orderStatusText": "<название состояния оплаты>" -- см. секцию 4.2.6
    "recurrentTemplateId": "<номер шаблона>",
    "refunds": [] -- список проведённых 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": {
    "orderNumber": "<номер оплаты>",
    "amount": "<сумма оплаты>",
    "merchant": "<номер торговой точки>",
    "terminal": "<номер терминала>",
    "userId": "<ID пользователя мерчанта>",
    "orderStatusCode": "<код состояния оплаты>", -- см. секцию 4.2.6
    "orderStatusText": "<название состояния оплаты>" -- см. секцию 4.2.6
    "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>",
      "dateTime": "<дата проведения>",
      "cardNumber": "<маскированный номер 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 4.2.6
    "orderStatusText": "<название состояния оплаты>" -- см. секцию 4.2.6
    "recurrentTemplateId": "номер шаблона платежа",
    "createdRecurrentTemplateId": "<номер шаблона рекуррентного платежа>"
    "refunds": [], -- список проведенных возвратов
    "transactions": [
      {dateTime:"дата проведения",
        "cardNumber": "",
        "amount": "сумма оплаты",
        "transactionId": "номер 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 |
| description | Описание заказа | Символьный, 1-255 Опциональный |
| email | Адрес электронной почты. Если для терминала настроено отображение поля «Email», то пользователю будет доступно изменение переданного значения. | Символьный, формат: [a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+ 1-255 Опциональный |
| phone | Номер телефона. Если для терминала настроено отображение поля «phone», то пользователю будет доступно изменение переданного значения. | Символьный формат: [0-9]{10} 10 символов Опциональный |

| | | |
|--------|--|---|
| userid | Уникальный идентификатор пользователя торговой точки. Может быть пустым (на стороне платежного шлюза не учитывается в логике). | Символьный, 1-50 Опциональный |
| sign | Подпись запроса | Символьный, 64 (для HMACSHA256) |

Успешный ответ содержит JSON-структуру:

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

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 Опциональный |
| recipientName | Наименование получателя денежных средств за услугу ЖКХ (юридическое лицо) | Строковый, 1-170 |
| recipientOkato | ОКАТО получателя денежных средств за услугу ЖКХ | Числовой, 8 Опциональный |
| yearOfPayment | 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 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 |

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 |

4.3 Алгоритм формирования HMAC

HMAC 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. HMAC формируется в два этапа: 1 – формирование строки данных для HMAC, 2 – формирование HMAC.

4.3.1 Подготовка строки данных для HMAC

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=1000000001
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=1000000001
terminal=1001
userid=101
```

Затем значения параметров преобразуются и складываются:

```
6100.0043https://example-merchant:8081/back-from-pay46Оплата за
электроэнергию3777111000000001410013101
```

Строка готова для генерации подписи.

4.3.2 Генерация НМАС

Для формирования НМАС используется алгоритм «**НМАСSHA256**». В качестве ключа используется секретный ключ терминала Торговой точки. Ключ представляет собой последовательность байт в HEX-формате, например:

```
b22ec899aaf398624c14305d56a3aa98095523fe
```

Но для формирования НМАС 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 НМАС 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=200460плата за электроэнергию3777111000000001410013101';  
$shared_key = 'b22ec899aaf398624c14305d56a3aa98095523ff';  
$hmac = hash_hmac('SHA256', $stringToSign, pack('H*', $shared_key));
```

Результат в \$hmac:

```
79c1947a8a9fced811af0a2f357aebdf027256761b926866eac65b4652323bcb
```

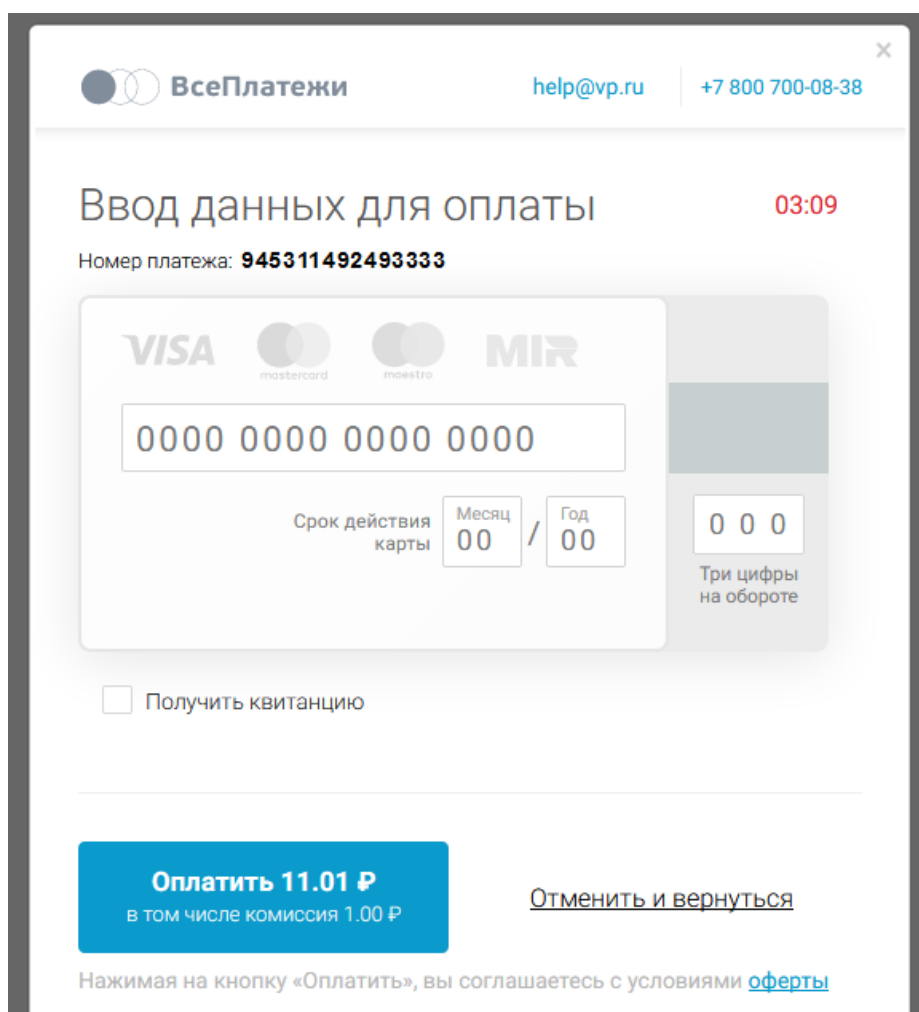
4.4.2 Java

```
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=200460плата за электроэнергию3777111000000001410013101';  
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;
```

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 точки 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 displays a payment widget from 'ВсеПлатежи'. At the top, there is a logo and contact information: 'help@vp.ru' and '+7 800 700-08-38'. The main heading is 'Ввод данных для оплаты' (Enter payment data) with a timer '03:09'. Below this, the payment ID is shown as 'Номер платежа: 945311492493333'. The card payment section includes logos for VISA, mastercard, maestro, and MIR. A card number field contains '0000 0000 0000 0000'. The expiration date is set to '00 / 00' for month and year. A CVV field contains '0 0 0' with the label 'Три цифры на обороте'. There is a checkbox for 'Получить квитанцию' (Get receipt). At the bottom, a blue button shows 'Оплатить 11.01 Р' (Pay 11.01 RUB) with a note 'в том числе комиссия 1.00 Р' (including 1.00 RUB commission). A link 'Отменить и вернуться' (Cancel and return) is also present. A footer note 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 инициирует 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"
    />
  </div>
</body>

```

```
<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 проведённых 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. <i>Формат: текстовый.</i> |
| Номер транзакции | Номер транзакции. <i>Формат: текстовый.</i> |

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

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 |