

Платёжный шлюз «ВсеПлатежи»

**Руководство Мерчанта
открытая передача ДДК**

Оглавление

История изменений.....	3
Раздел 1 Введение	4
Раздел 2 Определения	5
Раздел 3 Общие сведения	6
3.1 Защищённое соединение	6
3.2 Способы проведения оплаты	6
3.3 Уведомление об успешном проведении оплаты.....	6
Раздел 4 Описание API	8
4.1 Адреса серверов	8
4.2 Поддерживаемые запросы.....	8
4.2.1 Запрос на проведение оплаты	10
4.2.2 Запрос на проведение рекуррентного платежа.....	12
4.2.3 Запрос на блокировку средств	14
4.2.4 Запрос на списание заблокированных средств.....	16
4.2.5 Запрос на разблокировку средств	17
4.2.6 Запрос статуса заказа	18
4.2.7 Запрос расширенного статуса заказа.....	19
4.2.8 Запрос статуса заказа с информацией о транзакциях	21
4.2.9 Сообщение результатов 3ds	22
4.2.10 Коды и тексты состояний оплаты.....	23
4.3 Алгоритм формирования НМАС.....	23
4.3.1 Подготовка строки данных для НМАС.....	24
4.3.2 Генерация НМАС	24
4.4 Примеры для некоторых языков программирования.....	25
4.4.1 PHP.....	25
4.4.2 Java.....	25
Раздел 5 Коды ответов	26
Общие принципы	26
Коды ответов для запросов	26

История изменений

Версия	Дата	Изменения
1.0	09.08.2018	Документ создан на основе руководства мерчанта 3.3.
1.1	12.12.2018	Добавлена возможность создания и проведения рекуррентных платежей (автоплатежи): - Обновлён пункт 3.3 «Уведомление об успешном проведении оплаты» - Обновлён пункт 4.2 «Поддерживаемые запросы» (обновлен «Запрос на проведение платежа», добавлены «Запрос на проведение рекуррентного платежа», «Запрос расширенного статуса заказа»)
1.2	28.06.2019	Добавлены параметры для передачи телефона плательщика. Обновлены секции: 3.1 Защищённое соединение 4.2.1 Запрос на проведение оплаты 4.2.2 Запрос на проведение рекуррентного платежа 4.2.3 Запрос на блокировку средств 4.2.6 Запрос статуса заказа 4.2.7 Запрос расширенного статуса заказа Раздел 5 Коды ответов
1.3	19.05.2021	Добавлены новые параметры и обновлены разделы. Обновлены секции: 4.2.1 Запрос на проведение оплаты 4.2.3 Запрос на блокировку средств Добавлены разделы: Базовые параметры для 3dsv2 4.2.9 Сообщение cres от ACS
1.4	12.08.2021	Параметр RecurrentTemplateId изменён на createdRecurrentTemplateId в разделах: 3.3 Уведомление об успешном проведении оплаты 4.2.1 Запрос на проведение оплаты 4.2.3 Запрос на блокировку средств
1.5	25.08.2021	Добавлен новый код ошибки: ORDER_EXPIRED("239", "Заказ просрочен"), Секция: Коды ответов
1.6	16.09.2021	Обновлены секции: 4.2 Поддерживаемые запросы 4.2.1 Запрос на проведение оплаты 4.2.3 Запрос на блокировку средств 4.2.8 Запрос статуса заказа с информацией о транзакциях 4.2.9 Сообщение результатов 3ds
1.7	11.10.2021	Обновлена секция: Коды ответов

Раздел 1 Введение

Документ описывает порядок подключения к Платежному шлюзу «ВсеПлатежи» по протоколу передачи данных держателей карт.

Преимуществами такого подключения является возможность мерчанта настраивать внешний вид платежного шлюза на своем сайте.

Требования к мерчанту при подключении данному протоколу

1. Если у мерчанта уровень 1 по МПС, то он обязан иметь годный АОС (срок действия год с момента выдачи) с подписями мерчанта и аудитора в АОС + годный отчет по ASV сканированию (срок действия 3 месяца с момента получения).
2. Если у мерчанта уровень 2 или ниже по МПС, то он обязан иметь годный SAQ-D (срок действия год с момента выдачи) с подписями мерчанта и аудитора в SAQ-D + годный отчет по ASV сканированию (срок действия 3 месяца с момента получения).

Раздел 2 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.

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.

Раздел 3 Общие сведения

3.1 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.

3.2 Способы проведения оплаты

В данном руководстве рассмотрен только тип интеграции с открытой передачей данных держателя карты. Другие типы интеграции описаны в основном руководстве мерчанта шлюза.

3.3 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. createdRecurrentTemplateId
6. email
7. phone
8. 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 № **1000000023** (28.03.2017 13:56:10) на сумму **1500.00 руб.** успешно оплачен.

Описание заказа: Ежемесячный платёж по договору 0000992-11

Терминал: 80007001 Компания X города N

Подробную информацию по заказу вы можете посмотреть в личном кабинете.

--

С уважением,

Служба поддержки клиентов «ВсеПлатежи».

Раздел 4 Описание 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 с открытой передачей данных держателя карты	POST /api/pay	<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>
Запрос на проведение рекуррентного платежа через API с открытой передачей данных держателя карты	POST /api/recurrent	<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 проведение блокировки средств через API с открытой передачей данных держателя карты</p>	<p>POST /api/block</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 проведение списания заблокированных средств</p>	<p>POST /api/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</p>	<p>POST /api/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 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, соответствующем типу ошибки.</p>
<p>Запрос расширенного статуса заказа</p>	<p>POST /api/order/status-ext</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, соответствующем типу ошибки.</p>

Запрос статуса заказа с информацией о транзакциях	POST /api/order/status-v3	<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>
Сообщение результатов 3dsv1	POST /api/3dsresult	<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>

4.2.1 Запрос на проведение оплаты

Базовые параметры

URL	/api/pay
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
userIp	IP-адрес клиента, передавшего данные держателя карты	Символьный, Маска IPv4, через точку

recurrent	Флаг необходимости проведения автоплатежа	Символьный (TRUE/FALSE) Оptionальный
cardNumber	Номер карты	Числовой, 16-19
extMonth	Месяц окончания обслуживания карты (ММ)	Числовой, 2
extYear	Год окончания обслуживания карты (ГГ)	Числовой, 2
cvc2	Код безопасности карты	Числовой, 3-4
description	Описание заказа	Символьный, 1-255 Оptionальный
email	Адрес электронной почты.	Символьный, формат: [a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+ 1-255 Оptionальный
phone	Номер телефона.	Символьный формат: [0-9]{10} 10 символов 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
clientBackUrl	URL для возврата на сайт Торговой точки	Символьный, 1-255
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):	Символьный,

	screen.height	
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 Асепт из браузера пользователя	Символьный,
javaEnabled	Поддержка браузером пользователя Java	Символьный, (TRUE/FALSE) Опциональный
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

Успешный ответ содержит JSON-структуру:

```
{
  "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",
    "sign": "<Подпись ответа>",
    "terminal": "<Номер терминала>"
    "threeDSMethodURL": "<URL для 3ds>"
  }
}
```

Примечание: Если операция происходит с 3ds, то в ответе будет код rc 502 или 503, подробнее см. Сообщение результатов 3ds

4.2.2 Запрос на проведение рекуррентного платежа

Базовые параметры

URL	/api/recurrent
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
recurrentTemplateId	Номер шаблона для создания рекуррентного платежа. Возвращается при успешном создании автоплатежа в ответе на запрос проведения оплаты (см. секцию)	Символьный 1-255
description	Описание заказа	Символьный, 1-255 Опциональный
email	Адрес электронной почты.	Символьный, формат: [a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+ 1-255 Опциональный
phone	Номер телефона	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
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

Успешный ответ содержит JSON-структуру:

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

4.2.3 Запрос на блокировку средств**Базовые параметры**

URL	/api/block
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
userIp	IP-адрес клиента, передавшего данные держателя карты	Символьный, Маска IPv4, через точку
recurrent	Флаг необходимости проведения автоплатежа	Символьный (TRUE/FALSE) Опциональный
cardNumber	Номер карты	Числовой, 16-19
extMonth	Месяц окончания обслуживания карты (ММ)	Числовой, 2
extYear	Год окончания обслуживания карты (ГГ)	Числовой, 2

cvc2	Код безопасности карты	Числовой, 3-4
description	Описание заказа	Символьный, 1-255 Оptionальный
email	Адрес электронной почты.	Символьный, формат: [a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+ 1-255 Оptionальный
phone	Номер телефона	Символьный формат: [0-9]{10} 10 символов 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
clientBackUrl	URL для возврата на сайт Торговой точки	Символьный, 1-255
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 Ассерпт из браузера пользователя	Символьный,
javaEnabled	Поддержка браузером пользователя Java	Символьный (TRUE/FALSE) Опциональный
sign	Подпись запроса	Символьный, 64 (для HmacSHA256)

Успешный ответ содержит JSON-структуру:

```
{
  "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",
    "sign":"<Подпись ответа>",
    "terminal":"<Номер терминала>"
    "threeDSMethodURL":"<URL для 3ds>"
  }
}
```

Примечание: Если операция происходит с 3ds, то в ответе будет код rc 502 или 503, подробнее см. Сообщение результатов 3ds

4.2.4 Запрос на списание заблокированных средств

Базовые параметры

URL	/api/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:

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

4.2.5 Запрос на разблокировку средств**Базовые параметры**

URL	/api/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
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:

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

4.2.6 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 4.2.10
    "orderStatusText": "<название состояния оплаты>" -- см. секцию 4.2.10
    "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 4.2.10
    "orderStatusText": "<название состояния оплаты>" -- см. секцию 4.2.10
    "createdRecurrentTemplateId" : "<номер шаблона>",
    "refunds": [] -- список проведённых возвратов
    "email": "<адрес электронной почты>",
    "phone": "<номер телефона>".
  }
}
```

4.2.7 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 4.2.10
    "orderStatusText": "<название состояния оплаты>" -- см. секцию 4.2.10
    "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": {
    "orderNumber": "<номер оплаты>",
    "amount": "<сумма оплаты>",
    "merchantNumber": "<номер торговой точки>",
    "terminalNumber": "<номер терминала>",
    "userIdNumber": "<ID пользователя мерчанта>",
    "orderStatusCode": "<код состояния оплаты>", -- см. секцию 4.2.10
    "orderStatusText": "<название состояния оплаты>" -- см. секцию 4.2.10
    "recurrent": "true",
    "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>".
  }
}
```

4.2.8 Запрос статуса заказа с информацией о транзакциях

Базовые параметры

URL	/api/order/status-v3
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 4.2.10
    "orderStatusText": "<название состояния оплаты>" -- см. секцию 4.2.10
    "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>"
      "transactionStatusCode": "<код статуса транзакции>",
      "transactionStatusText": "<название статуса транзакции>",
      "iso": "<код ответа в соответствии ISO 8583>" - может отсутствовать
    }, {...} ] -- список всех транзакций
  }
}
```

* 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 4.2.10
    "orderStatusText": "<название состояния оплаты>" -- см. секцию 4.2.10
    "recurrent": "true",
    "createdRecurrentTemplateId": "<номер шаблона рекуррентного платежа>"
    "refunds": [], -- список проведённых возвратов
    "transactions": [
      {
        "dateTime": "дата проведения",
        "cardNumber": "",
        "amount": "сумма оплаты",
        "transactionId": "номер транзакции"
        "transactionStatusCode": "<код статуса транзакции>",
        "transactionStatusText": "<название статуса транзакции>",
        "iso": "<код ответа в соответствии ISO 8583>" -- может отсутствовать
      }
    ],
    "email": "<адрес электронной почты>",
    "phone": "<номер телефона>".
  }
}

```

4.2.9 Сообщение результатов 3ds

В случаях, когда необходимо пройти 3dsv1, в JSON ответе будет rc-код 502 и в теле json будут поля `acsurl`, `pareq`, `md`.

Далее Мерчанту необходимо перенаправить пользователя на страницу ввода 3ds (на адрес из `acsurl`), а потом отправить запрос с результатами Платежному шлюзу «ВсеПлатежи» на `/api/3dsresult`

В случаях, когда необходимо пройти 3dsv2, в JSON ответе будет rc-код 503 и в теле json будет поле `threeDSMethodURL`. Мерчанту необходимо направить пользователя на URL, который будет получен в параметре `threeDSMethodURL`. По окончании 3DS пользователь будет перенаправлен на `clientBackUrl` с передачей дополнительного параметра `result=0`, говорящего об успехе операции оплаты, или `result` не равного 0, в случае отказа в проведении оплаты.

Базовые параметры для 3dsv1

URL	<code>/api/3dsresult</code>
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

Параметры запроса

Название	Описание
<code>PaRes</code>	Параметр, передаваемый 3ds сервисом

MD	Параметр, передаваемый 3ds сервисом
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:

```
{
  "paramsMap":{
    "acsurl":"<Адрес для переадресации пользователя>",
    "pareq":"<значение pareq>",
    "md":"<идентификатор, полученный от банка>",
    "amount":"<Сумма заказа через точку>",
    "desc":"<Описание заказа>",
    "merchant":"<Номер мерчанта>",
    "orderId":"<Номер заказа>",
    "rc":"0",
    "sign":"<Подпись ответа>",
    "terminal":"<Номер терминала>"
  }
}
```

4.2.10 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. 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` → `13https://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.

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-pay460плата за
электроэнергию37771110000000001410013101
```

Строка готова для генерации подписи.

4.3.2 Генерация НМАС

Для формирования НМАС 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 НМАС он должен быть преобразован в бинарный массив. Данный ключ для наглядности можно представить в десятиричном 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 Коды ответов

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».

Коды ответов для запросов

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
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