

Инструкция по интеграции
Gateline
Платежный шлюз

Версия от 30.07.2024

Версии документа

Таблица 1. Версии документа

Дата	Предмет изменения
18.07.2024	Удален раздел Длинная запись
18.07.2024	В таблице 1.3 Сравнительная таблица способов интеграции удален пункт способа оплаты через GDS
18.07.2024	В таблице раздела 2.2 Статусы ордера обновлена запись по частичной отмене(reversal) и частичной оплате(refund)
18.07.2024	Удален раздел 4.2 Прямой процессинг , его переместили в общий раздел 4.2 Direct Api. Прямой процессинг
30.07.2024	Разделы и входящие в них подразделы 4.3 Прямой процессинг и 4.2. Direct API. Прямой процессинг поменяли местами.
18.07.2024	Добавлен раздел 4.2.3. Применение result3dsmethod
18.07.2024	В разделе 4.3.3.5 Отмена авторизации, возвраты Cancel добавлены сценарии по операциям частичной отмены и частичной оплаты
18.07.2024	В разделе 10 Проведение тестовых транзакций удалена информация по поведению тестового терминала в поле запроса Cardholder

Оглавление

1.	Способы интеграции.....	5
1.1.	Подключение через Simple API.....	5
1.2.	Подключение через API.....	5
1.3.	Сравнительная таблица способов интеграции	5
1.4.	Функциональные возможности платежного шлюза	7
1.4.1.	Антифрод-система	7
1.4.2.	Механизм 3D Secure	7
1.4.3.	Умный процессинг	7
1.4.4.	Установка лимитов	8
1.4.5.	Активация карт.....	8
1.4.6.	Запрос подтверждения оплаты	8
1.4.7.	Уведомления.....	9
1.4.8.	Потранзакционная сверка.....	9
1.4.9.	Проведение операций над платежами	9
1.4.10.	Система быстрых платежей Банка России	10
1.4.11.	Клиринг	10
1.4.12.	Платежная страница	11
1.4.12.1.	Форма в дизайне магазина	11
1.4.12.2.	Форма для мобильных устройств.....	11
1.4.12.3.	Выбор шаблона платежной страницы	11
1.4.13.	Личный кабинет партнера	12
2.	Организация ордеров и операций.....	13
2.1.	Общая информация.....	13
2.2.	Статусы ордера	13
2.3.	Статусы операций	14
3.	Работа через Simple API.....	15
3.1.	Оплата (POST /pay)	15
3.2.	Обработка результата оплаты	15
3.3.	Обработка ошибок.....	16
3.4.	Обработка чарджбеков	16
4.	Работа через API.....	17
4.1.	Схемы взаимодействия	17
4.2.	API. Процессинг через платежную форму GateLine	20

4.2.1.	Оплата (POST /checkout/pay).....	20
4.2.2.	Привязка карты (POST /checkout/activation).....	23
4.2.3.	Прочие методы	24
4.2.3.1.	Список ордеров (GET /order/)	24
4.2.3.2.	Список операций (GET /operation/)	25
4.2.3.3.	Статистика по кодам ответов (GET /statbymessage/)	26
4.2.3.4.	Информация об ордере (GET /order/:id).....	26
4.2.3.5.	Отмена авторизации, возвраты Cancel (POST /order/:id/cancel).....	27
4.2.3.6.	Списание денежных средств Settle (POST /order/:id/settle).....	28
4.2.3.7.	Применение Rebill (POST /order/:id/rebill)	29
4.2.3.8.	Применение Rebill3d (POST /order/:id/rebill3d).....	30
4.2.3.9.	Оплата через Google Pay (POST /order/googlepay)	30
4.2.3.10.	Оплата через Yandex Pay (POST /order/yapay)	32
4.2.3.11.	Оплата через Apple Pay (POST /order/applepay)	34
4.2.3.12.	Оплата через SberPay	36
4.2.3.13.	Тестовый запрос (GET /test/ping)	38
4.3.	Direct API. Прямой процессинг	38
4.3.1.	Применение Authorize (POST /order/authorize).....	38
4.3.2.	Применение Authorize3d (POST /order/authorize3d)	40
4.3.3.	Применение result3dsmethod (POST /order/:id/result3dsmethod).....	43
4.4.	Платежный виджет	44
5.	Уведомления.....	47
5.1.	Общая информация.....	47
5.2.	Требования к сайту, принимающему уведомления.....	47
5.3.	Контроль доставки уведомления.....	47
5.4.	Формат уведомления	47
5.5.	Верификация настроек уведомлений.....	47
5.6.	Активация	49
5.7.	Подтверждение оплаты	50
6.	Обработка редиректов.....	51
6.1.	Общая информация.....	51
6.2.	Проверка контрольной суммы	51
6.3.	Список передаваемых параметров.....	51
6.4.	Расшифровка статуса	52
7.	Обработка результата процессинга	53
8.	Работа с 3D Secure.....	54

8.1.	Общая информация	54
8.2.	Прямой процессинг	54
8.3.	Процессинг через платежную форму.....	55
8.4.	Активация и 3D Secure	55
8.5.	Обработка формы 3D Secure.....	55
9.	Клиринг	56
9.1.	Общая информация	56
9.2.	Автоматический режим.....	56
9.3.	Ручной режим	56
10.	Проведение тестовых транзакций	57
10.1.	Общая информация	57
10.2.	Использование 3D Secure.....	57
10.3.	Роли пользователей	57
11.	Формирование параметра extended.....	58

1. Способы интеграции

1.1. Подключение через Simple API

Подключение через Simple API позволяет работать только через платежную форму на стороне платежного шлюза, этот способ подходит для большинства типов электронной коммерции.

Simple API позволяет произвести базовую интеграцию и начать принимать платежи за несколько часов, при этом достаточно знания основ HTML.

При отправке запросов через Simple API в шлюз GateLine не требуется подписание запроса SSL сертификатом, а также редирект плательщика на платежную форму происходит автоматически.

Такая схема работы подходит как для небольших интернет-магазинов, которые продают единицы наименований и управляют выдачей товара в ручном режиме, так и для более крупных, где поддерживается корзина и автоматическая выдача товара при успешной оплате.

1.2. Подключение через API

Подключение через API занимает больше времени и требует более емкой интеграции Интернет Магазины с платежным шлюзом GateLine. Такой тип подключения стоит рассматривать, если предъявляется хотя бы одно из требований:

- Повышенные требования к безопасности;
- Интеграция с собственной системой учета продаж;
- Работа по схеме с привязкой карты или прямого процессинга;
- Работа с брендированной платежной формой как на стороне платежного шлюза, так и на стороне Интернет-магазина торгово-сервисного предприятия;
- Поддержка фискализации (ФЗ-54);
- Поддержка “длинной записи”
- Контроль над заказом (клиринг, отмена, возврат) средствами API;
- Работа с брендированной платежной формой.

1.3. Сравнительная таблица способов интеграции

Подключение	Simple API	API
Время подключения	1-4 часа	От 1-2 дня
Сложность подключения	Низкая	Высокая
Требуется сертификат для взаимодействия	Нет	Да
Основные возможности		
Антифрод	Есть	Есть

3D Secure	Есть	Есть
“Умный процессинг”	Только по странам	Полный
“Длинная запись”	Нет	Есть
Установка лимитов	Есть	Есть
Активация карт	Есть	Есть
Запрос подтверждения оплаты	Есть	Есть
Уведомления	E-mail, HTTP	HTTP
Потранзакционная сверка	Через веб-интерфейс	Через веб-интерфейс и API
Проведение операций над платежами	Веб-интерфейс	Веб-интерфейс, API
Клиринг	Автоматический, отложенный, ручной (веб-интерфейс)	Автоматический, отложенный, ручной (API, веб-интерфейс)
Схемы работы	Платежная страница	Платежная страница, прямой процессинг, процессинг с привязкой
Оплата через СБП	Есть	Есть
Поддержка 54-ФЗ	Есть	Есть
Платежная страница		
Индивидуальный дизайн	Есть	
Мультиязычность	Есть	
Поддержка мобильных устройств	Есть	
Веб-интерфейс		
Разделение доступа	Есть	
Управление заказами	Отмена, возврат, частичный возврат, списание	
Статистика продаж	Есть	

Аналитика	Есть
Проекты	Есть
Стоп-листы	Есть
Мониторинг состояния	Есть

1.4. Функциональные возможности платежного шлюза

1.4.1. Антифрод-система

Система использует два механизма Антифрод-защиты:

- Скоринговая система, которая оценивает риск проведения транзакции по каждому заказу и реализовать определенные алгоритмы по обработке данных заказов;
- Механизм стоп-листов (отклонение транзакций по номеру карты, IP-адресу и другим параметрам).

1.4.2. Механизм 3D Secure

Главный современный инструмент для обеспечения безопасного приема платежей и снижения количества мошеннических транзакций – это механизм 3D Secure (MasterCard SecureCode и Verified by Visa).

Совершая платеж, клиент направляется на сайт банка, выпустившего карту, где производится его аутентификация. Под этим обычно подразумевается ввод платежного пароля или одноразового пароля из SMS-уведомления, но реализация может зависеть от банка-эмитента. Таким образом удостоверяется, что картой пользуется ее владелец, а не злоумышленник.

3D Secure позволяет не только уменьшить риск фрода, но и во многих случаях позволяет перенести ответственность за чарджбеки с магазина на банк-эмитент.

1.4.3. Умный процессинг

Несмотря на все преимущества 3D Secure, этот механизм усложняет процедуру оплаты для клиента и может служить причиной значительного снижения количества успешных платежей. Оптимальной стратегией является использование 3Ds не для всех платежей, а только для входящих в группу повышенного риска.

Шлюз позволяет гибко управлять тем, какие платежи направлять в 3D Secure, например,

- Карта эмитирована не российским банком;
- Сумма операции превышает 100 тыс. рублей;
- Карта MasterCard, выпущена австралийским банком.

Возможности настройки не ограничены вышеперечисленными примерами и могут подбираться индивидуально для каждого партнера, в зависимости от специфики его бизнеса.

Помимо этого, шлюз предоставляет возможность магазину самостоятельно решать, использовать ли 3D Secure для каждого платежа отдельно, например, отключать 3D Secure для постоянных клиентов.

Некоторые банки выпускают карты, платежи по которым возможны только через 3DS. Если клиент пытается совершить платеж по такой карте обычным образом, операция отклоняется, и продажа не происходит. Шлюз имеет возможность обрабатывать такие случаи. Во-первых, во многих случаях по номеру карты можно определить ее свойства, и принудительно направить ее на процессинг с 3Ds. Во-вторых, даже если попытка обычной авторизации не удалась, система попытается провести такую же операцию через 3D Secure. Это позволяет успешно провести платеж по таким картам, тогда как в обычных условиях оплата бы завершилась неудачей уже после первой попытки.

1.4.4. Установка лимитов

С помощью системы настраиваемых правил, шлюз может автоматически следить за потоком операций и не допускать превышения установленных безопасных лимитов. Правила могут выглядеть примерно так:

- Максимальная сумма одной операции: 30 000 руб.
- Максимальное количество платежей по одной карте в сутки: 5
- Максимальная сумма платежей по одной карте в сутки: 100 000 руб.

Система поддерживает множество различных правил, которые могут настраиваться по нескольким параметрам и комбинироваться друг с другом.

Подобный набор правил можно установить, как на магазин в целом, так и на каждую торговую точку, открытую для магазина в банке.

Хорошей альтернативой лимитам, срабатывание которых ведет к отклонению транзакции, является применение “Умного процессинга”. В этом случае транзакции сверх установленных лимитов не отклоняются, а направляются в 3D Secure.

1.4.5. Активация карт

Когда покупатель впервые оплачивает покупку с помощью карты, шлюз предлагает пройти простую дополнительную процедуру аутентификации: на карте блокируется небольшая случайная сумма, после чего клиенту предлагается указать списанную сумму с точностью до копеек либо код авторизации. Покупатель может узнать эту информацию с помощью SMS уведомления от своего банка, через интернет-банк, либо через обращение в банк по телефону, таким образом удостоверив, что он является владельцем карты.

Процедура активации встроена в платежную страницу таким образом, что процесс оплаты не прерывается и от клиента не требуется несколько раз вводить платежные данные.

Активация не может являться полноценной заменой 3D Secure, поскольку процедура аутентификации запрашивается однократно при первой оплате. Однако ее можно рассматривать как хорошее компромиссное решение для повышения безопасности платежей для тех случаев, когда 3D Secure недоступен.

1.4.6. Запрос подтверждения оплаты

Когда покупатель попал на платежную страницу, может случиться так, что он будет вводить данные дольше, чем ожидает магазин, и за это время актуальность его платежа может потеряться. Например, это может произойти при покупке ж/д-билета, бронь которого сохраняется ограниченное

время, или при покупке последнего доступного на складе товара, который может быть перехвачен другим покупателем.

Чтобы решить эту проблему, шлюз может запрашивать у магазина, следует ли проводить платеж непосредственно перед оплатой. Если магазин ответит отрицательно, оплата будет отклонена и клиент будет перенаправлен обратно в магазин.

1.4.7. Уведомления

Как правило, магазин должен иметь возможность своевременно получать информацию о том, успешным или неуспешным был платеж, чтобы принять решение о выдаче товара покупателю. После оплаты шлюз перенаправляет пользователя в магазин, прилагая информацию о результате платежа, но этот способ нельзя рассматривать как надежный, т.к. пользователь может отказаться от перенаправления или вовсе закрыть окно браузера после оплаты.

Поэтому шлюз дополнительно отправляет асинхронные уведомления о совершенных платежах в режиме реального времени по электронной почте или по HTTP/HTTPS.

Уведомления по e-mail подходят для небольших магазинов, которые могут обрабатывать такие уведомления в ручном режиме.

HTTP/HTTPS-уведомления требуется принимать в автоматическом режиме, этот способ позволяет автоматически помечать заказ как оплаченный и выдавать покупателю товар.

Уведомления могут отправляться не только при совершении платежа (авторизации), но и при проведении всех прочих операций: отмены, возврата, фин. подтверждения платежа и т.д. Таким образом, магазин может поддерживать финансовую синхронизацию своей базы заказов.

1.4.8. Потранзакционная сверка

Магазин может регулярно (ежедневно) сверять список транзакций, проведенных через шлюз, чтобы исключить расхождение сумм платежей или такой ситуации, когда в шлюзе платеж помечен как ошибочный, а в магазине – как успешный и т.п.

Такую сверку можно организовать как в автоматическом, так и в ручном режиме. Шлюз позволяет магазину автоматически выгружать список операций через API, при этом поддерживается фильтрация по дате, по типу и успешности операции.

Через веб-интерфейс можно выгрузить список операций в формате XML, CSV или Excel. В выгружаемых данных присутствует вся необходимая информация для сверки: тип и статус операции, сумма, дата проведения и т.д.

1.4.9. Проведение операций над платежами

Шлюз предоставляет возможность управлять заказами после блокировки суммы на карте клиента. Поддерживаются следующие операции:

- отмена авторизации;
- финансовое подтверждение (клиринг), в том числе частичный;
- возврат, в том числе частичный.

Операции могут проводиться как через веб-интерфейс, так и через API. Проведение операций через API обычно требуется, когда магазину необходимо подключить к шлюзу свою систему учета продаж.

1.4.10. Система быстрых платежей Банка России

Через систему быстрых платежей (СБП) Банка России Покупатель – физическое лицо может оплатить покупку или услугу, предоставленную юридическим лицом – торгово-сервисным предприятием (ТСП) со своего банковского счета в Банке Покупателя на расчетный счет ТСП в Банке ТСП. При этом, Банк Покупателя и Банк ТСП должны быть участниками СБП. Заплатить через СБП может Покупатель, у которого есть мобильное приложение Банка Покупателя.

С точки зрения интеграции для предоставления Покупателю возможности оплаты товаров/услуг через СБП существуют два варианта взаимодействия ТСП с Платежным шлюзом GateLine:

- ТСП при взаимодействии с платежным шлюзом GateLine используются те же методы проведения оплаты, как по банковским картам, QR-код отображается на отдельной вкладке платежной формы платежного шлюза GateLine. Предварительно такой метод оплаты должен быть подключен к платежной форме администратором Платежного шлюза GateLine.
- Ссылка на QR-код передается в ответе при использовании в API GateLine метода оплаты POST /checkout/pay ([п. 4.2.2.](#)). ТСП самостоятельно отображает QR-код Покупателю.

При этом в уведомление о проведенной операции в поле ответа «payment_type» указывается тип инструмента оплаты (п.5.5.)

1.4.11. Клиринг

Когда клиент проводит оплату, сумма платежа сначала блокируется на карте, при этом фактически средства не списываются до тех пор, пока не будет проведено финансовое подтверждение. Пока подтверждение не произошло, сумму можно разблокировать и никаких движений денежных средств на карте клиента не произойдет. После подтверждения сумма платежа списывается с карты клиента, а магазин несет издержки, связанные с проведением этой операции.

Для большинства магазинов подходит автоматический отложенный клиринг. В этом случае в момент оплаты сумма платежа блокируется (авторизуется) на карте клиента, а через некоторое время после этого (например, через 6 часов) система автоматически списывает заблокированную сумму. Если в магазине часто случаются возвраты/отмены платежей, полезно выбрать такую задержку, чтобы большинство отмен происходили до проведения списания. В этом случае сумма платежа просто разблокируется, и магазин не понесет издержек, связанных с проведением этой транзакции.

Если не использовать автоматический клиринг, средства на карте могут быть заблокированы на срок до 45 дней, в зависимости от банка-эквайера. Это возможность полезна, например, для компаний, работающих в сфере бронирования гостиничных номеров: в момент оплаты сумма блокируется на карте клиента, а списание производится при фактическом заселении клиента.

Доступные варианты настройки клиринга:

- Отложенный: списание проводится с задержкой от 1 часа до 45 дней после авторизации.
- Автоматический: средства списываются с карты непосредственно при оплате
- Ручной: запрашивается в произвольный момент через API или веб-интерфейс.

1.4.12. Платежная страница

Удобство и доступность платежной страницы напрямую влияет на процент успешных продаж. По умолчанию шлюз предоставляет платежную форму, оформленную в своем стиле, она обладает такими достоинствами:

- Автоматическое отображение упрощенной версии формы для мобильных устройств.
- Все действия происходят без перезагрузки страницы. Если процедура оплаты состоит из нескольких шагов, клиенту не требуется несколько раз вводить данные своей карты.
- Отображение результата операции на стороне магазина: шлюз всегда направляет клиента в магазин, передавая результат платежа и причину ошибки, если платеж не завершился успехом.

1.4.12.1. Форма в дизайне магазина

Магазин может предоставить свой вариант формы, которая будет иметь такой внешний вид, к которому привыкли клиенты. Шлюз берет на себя техническую интеграцию формы магазина, это включает в себя клиентскую валидацию, защиту от повторного нажатия кнопки “Оплатить” и т.д. В то же время форма магазина может содержать любые элементы, ссылки, клиентские скрипты.

Брендируемая форма может быть представлена в двух вариантах: обычном и для мобильных устройств. К ним предъявляются общие технические требования. Если мобильная форма не предоставлена, на мобильных устройствах может отображаться как полноценная брендируемая форма магазина, так и стандартная форма для мобильных устройств, предоставляемая шлюзом.

1.4.12.2. Форма для мобильных устройств

Система может автоматически определить, что клиент открывает платежную форму с помощью мобильного телефона, планшета и т.д. В этом случае открывается вариант формы для мобильных устройств, вид которой оптимизирован для отображения на небольшом экране. Также на мобильной форме не применяются клиентские скрипты, которые могут не поддерживаться некоторыми разновидностями мобильных браузеров, поэтому форма максимально совместима с разными типами мобильных устройств.

Магазин может управлять тем, какая форма отобразится каждому клиенту: обычная или мобильная. Если тип формы явно не указан, шлюз определит его автоматически.

Магазин может предоставить свой вариант мобильной формы. Это форма, во-первых, может быть выполнена в дизайне магазина, и во-вторых, она может быть оптимизирована под какое-то конкретное устройство, которым пользуются клиенты.

1.4.12.3. Выбор шаблона платежной страницы

В настройках сайта указывается, какие шаблоны используются по умолчанию для обычных и мобильных устройств.

Магазин может принудительно выбрать шаблон для платежа, указав имя шаблона в запросе на получение уникальной ссылки на платежную страницу. Список доступных шаблонов требуется получить у службы поддержки.

Если для платежа шаблон выбран принудительно, он будет использован, независимо от устройства пользователя, поэтому магазин должен учитывать тип устройства пользователя при выборе.

1.4.13. Личный кабинет партнера

Заказы в личном кабинете(ЛК) представлены в виде списка в обратном хронологическом порядке. С помощью системы фильтров можно отобрать заказы по нужному критерию, например, все успешные за определенный день.

По каждому заказу можно просмотреть расширенную информацию, которая включает платежные данные клиента, историю проведения операций и т.д.

Для того, чтобы предоставить доступ к ЛК разным сотрудникам, можно заводить субпользователей с разными правами доступа. Например, можно добавить пользователя с правами “Служба поддержки”, который будет иметь доступ к просмотру информации о платежах, и оператора, который дополнительно получит право проводить операции отмены.

Статистика продаж формируется в режиме реального времени, она показывает количество транзакций, оборот и прочие данные, просуммированные по дням или по другим параметрам. Помимо этого, вычисляется сумма удержанных комиссий, возвращенных средств и т.п.

Аналитика позволяет оценивать динамику оборота, с ее помощью можно узнать, в какие дни недели происходит больше всего продаж, или в какое время суток активность покупателей наибольшая.

Партнер может разделять все платежи своего сайта на несколько групп (проекты), с тем чтобы оценивать статистику независимо по каждому проекту. Например, если магазин продает товары двух принципиально разных групп (авиабилеты и бронирование гостиниц), имеет смысл проводить их по разным проектам.

Если был обнаружен платеж по карте, который признается мошенническим, эту карту можно добавить в стоп-лист, так чтобы в дальнейшем платежи по ней были отклонены. В случае необходимости, карту из стоп-листа можно исключить, тогда транзакции по ней вновь станут возможны.

Единая панель состояния системы оценивать количество различных ошибок, произошедших за текущий день или другой выбранный период. Регулярный контроль этой страницы поможет быстро обнаружить проблемы, если они возникнут. Параметр “Проходимость системы” можно использовать для оценки эффективности настройки “Умного процессинга”.

2. Организация ордеров и операций

2.1. Общая информация

В отличие от традиционной схемы транзакций, когда каждая операция содержит в себе все относящиеся к ней данные, в системе реализована схема заказов и связанных с ними операций.

Заказ выступает хранилищем информации о клиенте: номер карты, имя держателя карты, биллинг-адрес. Кроме этого, заказ несет информацию о текущем состоянии процессинга, дате проведения последней операции и т.д.

Создание заказа происходит в момент проведения первой операции, обычно это authorize. В ответе всегда передается уникальный идентификатор заказа (order_id). В дальнейшем полученный идентификатор используется для проведения всех операций с этим заказом, он является обязательным параметром во всех запросах в API, связанных с проведением операций.

2.2. Статусы ордера

По статусу ордера можно судить о текущем состоянии заказа.

Статус	Расшифровка	Возможные операции
initial	Ордер создан, но процессинг еще не начался.	–
processing	Выполняется процессинг операции	–
authorize	Проведена успешная авторизация (блокировка суммы)	settle cancel rebill(3d)
prepare3d	Запущена процедура аутентификации через 3D Secure	–
sale	Проведена успешная операция settle (списание суммы)	cancel chargeback rebill(3d)
reversal	Произведена частичная или полная отмена	rebill(3d)
refund	Произведен частичный или полный возврат средств	refund chargeback rebill(3d)
decline	Авторизация отклонена (не удалось заблокировать сумму)	rebill(3d)
error	В процессе выполнения операции произошла ошибка, возможно, требуется вмешательство службы поддержки.	–
chargeback	Был произведен чарджбек (сумма возвращена по инициативе клиента).	–
ruledcline	Авторизация была отклонена в соответствии с правилами сайта	–

2.3. Статусы операций

По статусу операции можно судить о результате ее выполнения.

Значение	Описание
success	Операция проведена успешно.
failed	Был получен ответ от банка, в котором указано, что операция не завершилась удачно по какой-либо причине (например, произошел отказ в авторизации).
processing	Операция в процессе выполнения
error	Возникла проблема, при которой ответ от банка не был получен (например, из-за ошибки подключения к банку или внутренней ошибки системы).

3. Работа через Simple API

Общая информация

- Оплата инициируется отправкой клиентом формы на специальный URL в системе
- Тип кодировки: application/x-www-form-urlencoded
- Метод формы: POST
- Магазин формирует форму на своей стороне и отображает ее клиенту

Подписывание формы

Подпись представляет собой HMAC-SHA1-сумму строки, которая включает в себя список всех параметров (имя, значение). Параметры сортируются по ключу в алфавитном порядке.

Сумма вычисляется с паролем сайта.

Составляющие подписи разделяются символом “;” (точка с запятой). Пример

формирования подписи:

```
HMAC_SHA1 ("param1=value1;param2=value2", mypassword)
```

Где mypassword - пароль сайта.

Подпись передается параметром checksum в этой же форме.

3.1. Оплата (POST /pay)

Параметры формы:

Имя	Описание	Пример
amount	Сумма операции	100.99
description	Описание заказа	Товар (#123456789)
site	Идентификатор сайта	mysite
email *	E-mail клиента	mail@example.com
merchant_order_id *	Идентификатор заказа в системе клиента	12345-NB
locale *	Язык платежной формы en, ru	en
checksum	Подпись формы	e3669...6319da

* - опциональные поля

При успешной обработке формы клиент перенаправляется (HTTP 302 Redirect) на платежную форму. В случае ошибок клиенту будет сгенерирована страница с описанием ошибок.

3.2. Обработка результата оплаты

Если оплата прошла успешно, система отправляет уведомление об успешной операции. Магазин может автоматически принимать уведомление и основываясь на нем принимать решение об отгрузке товара. Подробнее о принципах обработки уведомлений.

Клиент после оплаты перенаправляется в магазин.

3.3.Обработка ошибок

Сообщения об ошибках отображаются клиенту на стороне шлюза. Возможные ошибки:

- Forbidden – режим Simple API не разрешен для данного сайта.
- Amount/Description/Site required – не все обязательные поля заполнены.
- Checksum invalid – неправильная подпись.

3.4.Обработка чарджбеков

Чарджбек является операцией, которая проводится только по инициативе системы.

При проведении операции доставляется уведомление. Формат уведомления

соответствует формату для любой другой операции.

Передаваемые поля:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d5151807781360baf9
merchant_order_id	Идентификатор заказа в системе клиента	123456789
message	Описание результата	Success
status	Статус операции	success
time	Дата выполнения запроса	2011-05-31 12:13:15

4. Работа через API

4.1. Схемы взаимодействия

В зависимости от количества платежей, от того на чьей стороне выполняются операции по приему и обработке данных держателей карт (ДДК) или обработка платежных данных карт (ПДК) Продавцу и Платежному шлюзу требуется подтверждение соответствия PCI DSS, без которого работать запрещено. Минимальные требования по соответствию стандарта PCI DSS будут предъявляться к Продавцу в случае проведения всех операций с ДДК и ПДК на стороне Платежного шлюза (лист самооценки SAQ A). Платежный шлюз, имеет сертификат о подтверждении соответствия на прием и обработку ДДК и ПДК, без которого работать запрещено. Продавцу, в случае выбора варианта самостоятельного приема и обработки ДДК и ПДК необходимо так же самостоятельно подтвердить соответствие PCI DSS на данные операции (более высокие требования лист самооценки SAQ A-EP или аудит PCI DSS).

Процессинг через платежную форму

1. Пользователь нажимает кнопку “Оплатить” на сайте продавца.
2. Продавец обрабатывает запрос пользователя на совершение покупки и отправляет в систему запрос рау, указывая описание товара, сумму платежа и при необходимости другие данные.
3. В ответ система передает продавцу URI платежной формы в системе.
4. Продавец перенаправляет пользователя на полученный URI.
5. Пользователь вводит платежную информацию в форму.
6. Если затребован процессинг только активированных карт:
 - a. Система обрабатывает форму и проверяет, является ли введенный номер карты активированным. Если карта уже была активирована ранее, переходим к шагу 7.
 - b. Система предупреждает пользователя, что необходима процедура активации, и после его согласия блокирует небольшую случайную сумму на карте.
 - c. Пользователь узнает в своем банке списанную сумму или код авторизации и вводит значение в форме подтверждения активации. Если оно введено верно, система предлагает завершить процедуру оплаты и отображает кнопку “Оплатить”.
 - d. Если активация была проведена успешно, система отправляет уведомление, которое содержит информацию о проведенной операции activation.
7. Система обрабатывает форму и перенаправляет пользователя в магазин продавца.
8. Если платеж был проведен успешно, система отправляет уведомление, которое содержит информацию об авторизации.
9. Продавец обрабатывает результат проведения операции и формирует страницу-результат.

Процессинг с привязкой карты

Во время работы по данному алгоритму может потребоваться передача параметра CVV в разрезе операций rebill (Зависит от настроек терминала, выданного банком). Данная операция означает самостоятельную обработку платежных данных карт, на которые Продавцу требуется подтверждение соответствия PCI DSS, иначе передавать ПДК запрещено.

Привязка карты:

1. Пользователь нажимает кнопку “Привязать карту” на сайте продавца.

2. Продавец обрабатывает запрос пользователя на привязку карты и отправляет в систему запрос activation.
3. В ответ система передает URI, который ведет пользователя на форму активации в системе.
4. Продавец перенаправляет пользователя на полученный URI.
5. Пользователь вводит платежную информацию в форму активации.
6. Система блокирует небольшую случайную сумму на карте пользователя.
7. Пользователь узнает код авторизации или заблокированную сумму и вводит в форму подтверждения активации в системе.
8. Система перенаправляет пользователя в магазин продавца.
9. Если активация была проведена успешно, система отправляет уведомление, которое содержит информацию о проведенной активации.
10. Продавец обрабатывает уведомление и отображает пользователю информацию о результате привязки.
11. Продавец сохраняет идентификатор (order_id), который будет использован для проведения платежей по привязанной карте.

Процессинг:

1. Пользователь нажимает кнопку “Оплатить” на сайте продавца.
2. Продавец обрабатывает запрос пользователя на совершение покупки и отправляет в систему запрос rebill, указывая идентификатор (order_id), сохраненный во время привязки карты.
3. В ответ система передает результат проведения операции.
4. Если платеж был проведен успешно, система отправляет уведомление, которое содержит информацию о проведенной операции.
5. Продавец обрабатывает результат проведения операции и формирует страницу-результат.

Прямой процессинг

Так как в ходе данной интеграции используется собственная платежная форма торгового предприятия, это подразумевает хранение и передачу карточные данные (ДДК) плательщиков. На самостоятельную обработку и передачу ДДК. Продавцу необходимо соответствующее подтверждение PCI DSS.

1. Пользователь нажимает кнопку “Оплатить” на сайте продавца.
2. Продавец обрабатывает запрос пользователя на совершение покупки и отправляет в систему запрос authorize.
3. В ответ система передает результат проведения операции.
4. Если платеж был проведен успешно, система отправляет уведомление, которое содержит информацию о проведенной операции.
5. Продавец обрабатывает результат проведения операции и формирует страницу-результат.

Общая информация

- Запросы осуществляются по HTTP 1.1 с использованием SSL и клиентского сертификата.
- Кодировка запросов и ответов: UTF8

- Ответы на запросы в API даются в формате XML.
- Для выгрузки информации используется метод GET, для проведения манипуляций над данными – метод POST.
- Каждый запрос подписывается с использованием секретного ключа сайта, принципы формирования подписи для POST и GET запросов описаны ниже.
- Для GET-запросов параметры запроса передаются как QUERY STRING
- Для POST-запросов параметры передаются в теле POST-запроса (application/x-www-form-urlencoded).
- Успешные ответы отдаются с HTTP-статусом 200. В случае ошибки статус может отличаться.

Подписывание запроса

Подпись представляет собой HMAC-SHA1-сумму строки, которая включает в себя:

- Путь (например, /operation)
- Список всех параметров (имя, значение). Параметры сортируются по ключу в алфавитном порядке.

Составляющие подписи объединяются символом “;” (точка с запятой). Пустой список параметров интерпретируется как пустая строка, таким образом точка с запятой должна присутствовать в любом случае. Все методы указываются без “/” слеша.

Сумма вычисляется с паролем сайта.

Пример формирования подписи:

```
HMAC_SHA1 ("/pay;param1=value1;param2=value2", mypassword)
```

Подпись вместе с идентификатором сайта передается в заголовке X-Authorization HTTP-запроса, например:

```
X-Authorization: mysite 57bf95da3daf0ac9707df969cc935405
```

Обработка ошибок

Сообщения об ошибках передаются в едином формате:

```
<?xml version="1.0" encoding="utf-8"?>
<error>
  <message>An error occurred</message>
</error>
```

Расшифровка ошибки находится в поле message.

Дополнительную информацию несет HTTP-код ответа:

Код	Описание	Рекомендация
400	Неправильный запрос	Исправить ошибки и повторить запрос
403	Доступ запрещен	–
404	Ресурс не найден	Убедиться в правильности запрашиваемого

		URL
406	Операция отклонена из-за срабатывания одного или более правил-ограничений сайта или терминала	–
500	Системная ошибка	Обратиться в службу поддержки для получения дополнительной информации.
504	Срабатывание таймаута при подключении к банку.	Повторить запрос.

Выборка данных

При выгрузке данных, таких как, например, список ордеров, максимальное количество записей, которые можно получить за один запрос, составляет 2000 записей. Рекомендуется использовать параметры фильтрации, чтобы ограничить количество выгружаемых данных.

4.2. API. Процессинг через платежную форму GateLine

4.2.1. Оплата (POST /checkout/pay)

- Пользователь нажимает кнопку “Оплатить” на сайте продавца.
- Продавец обрабатывает запрос пользователя на совершение покупки и отправляет в систему запрос pay, указывая описание товара, сумму платежа и при необходимости другие данные.
- В ответ система передает продавцу URI платежной формы в системе.
- Продавец перенаправляет пользователя на полученный URI.
- Пользователь вводит платежную информацию в форму.
- Если затребован процессинг только активированных карт:
- Система обрабатывает форму и проверяет, является ли введенный номер карты активированным. Если карта уже была активирована ранее, переходим к шагу 7.
- Система предупреждает пользователя, что необходима процедура активации, и после его согласия блокирует небольшую случайную сумму на карте.
- Пользователь узнает в своем банке списанную сумму или код авторизации и вводит значение в форме подтверждения активации. Если оно введено верно, система предлагает завершить процедуру оплаты и отображает кнопку “Оплатить”.
- Если активация была проведена успешно, система отправляет уведомление, которое содержит информацию о проведенной операции activation.
- Система обрабатывает форму и перенаправляет пользователя в магазин продавца.
- Если платеж был проведен успешно, система отправляет уведомление, которое содержит информацию об авторизации.
- Продавец обрабатывает результат проведения операции и формирует страницу-результат.

При запросах в API для процессинга через форму оплаты в успешном ответе всегда передается поле redirect, которое содержит URI, на который необходимо перенаправить пользователя.

Если на этапе обработки запроса произошла ошибка, ответ соответствует стандартному ответу с ошибкой.

Принимаемые параметры:

Имя	Описание	Пример
merchant_order_id	Идентификатор заказа в системе клиента	123456789
amount	Сумма платежа, отображается на платежной форме	100.99
description	Описание платежа, отображается на платежной форме. HTML-разметка не поддерживается.	Электронный товар (заказ #123456789)
force3d *	см. Работа с 3D Secure	1
return_success_url *	URL для перенаправления клиента в случае успешной операции должен быть передан согласно стандарту RFC 3986	http://example.com/success?
return_failure_url *	URL для перенаправления клиента в случае ошибки должен быть передан согласно стандарту RFC 3986	http://example.com/failure?
activation_required *	Необходима ли активация карты. Если передано, значение перекрывает соответствующую настройку сайта (“Только активированные карты”).	1
email *	Е-mail клиента	mail@example.com
project *	Название проекта	promosite
currency *	Цифровой код валюты согласно ISO 4217	643
exchange_rate *	Курс валюты относительно рубля	0.05
currency_amount *	Сумма заказа отображаемая на платежной форме в случаи если currency не равно 643	123
phone *	Номер телефона клиента (до 20 символов)	+74951231212
custom_numeric *	Числовое пользовательское поле (8 байт)	123456789
custom_text *	Текстовое пользовательское поле (до	12345-NB-123

	12 символов)	
terminal *	Дескриптор терминала	MYSHOP.RU
timeout *	Время жизни токена платежной формы в минутах или секундах.	15 (15 минут) 831s (831 секунда)
mobile	Использовать специальную версию платежной страницы для мобильных устройств	auto
locale *	Язык платежной формы en, ru	en
paysystem *	Параметр, который регистрирует ссылки на оплату в указанных платежных системах и возвращает их в дополнительных параметрах ответа (можно передать несколько)	sberpay,sbp
extended *	Дополнительные параметры	См. 11. Формирование параметра extended

* - Опциональные поля

Возможные значения параметра mobile:

Значение	Описание
auto (по умолчанию)	Показывать мобильную версию только для мобильных устройств, устройство определяется автоматически по User-Agent
off	Всегда показывать полную версию
on	Всегда показывать версию для мобильных устройств

Поля ответа:

Имя	Описание	Пример
redirect	URL платежной формы	https://checkout.sandbox.gate line.net:18310/pay?token=***
qrurl*	URL на оплату через СБП НСПК. Ссылка использует deeplink для связи с мобильными приложениями банка для оплаты. Ссылка может быть закодирована в QR-код для оплаты путем сканирования мобильным устройством.	https://qr.nspk.ru/****?type=02&bank=100000000005&sum=300&cur=RU B&crc=8B46

* - Опциональные поля

Для получения поля "qrurl" необходимо в поле "extended" передать параметр {"qr_sbp":1} (см. 4.1.3.5 Формирование параметра extended)

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<checkout>
  <тип_запроса>
    <redirect>https://checkout.example.com/checkout/pay? token=abc</redirect>
  </тип_запроса>
</checkout>
```

4.2.2. Привязка карты (POST /checkout/activation)

Данный метод предназначен для привязки карты на стороне ПШ, для выполнения регулярных платежей. В данном случае после успешной активации карты необходимо сохранить order_id для передачи его в методе Rebill.

Принимаемые параметры:

Имя	Описание	Пример
merchant_order_id	Идентификатор заказа в системе клиента	123456789
try3d *	см. Работа с 3D Secure	1
return_success_url *	URL для перенаправления клиента в случае успешной операции должен быть передан согласно стандарту RFC 3986	http://example.com/success?
return_failure_url *	URL для перенаправления клиента в случае ошибки должен быть передан согласно стандарту RFC 3986	http://example.com/failure?
email *	Е-mail клиента	mail@example.com
phone *	Номер телефона клиента (до 20 символов)	+74951231212
custom_numeric *	Числовое пользовательское поле (8 байт)	123456789
custom_text *	Текстовое пользовательское поле (до 12 символов)	12345-NB-123
terminal *	Дескриптор терминала	MYSHOP.RU
mobile	Использовать специальную версию платежной страницы для мобильных устройств	no
extended *	Дополнительные параметры	См. 11. Формирование параметра extended

* - Опциональные поля

4.2.3. Прочие методы

Следующие методы необходимы для управления заказами или получения их статуса.

4.2.3.1. Список заказов (GET /order/)

Принимаемые параметры:

Имя	Описание	Пример
order_created_from	Дата создания заказа (от)	2011-05-01
order_created_to	Дата создания заказа (до)	2011-05-31
order_status	Статус	authorize
order_client_address	Адрес	123 Main Street
order_client_cardholder	Имя держателя карты	Ivan Ivanov
order_client_city	Город	Springfield
order_client_country	Страна	US
order_client_email	E-mail	mail@example.com
order_client_ipaddr	IP-адрес клиента	123.123.123.123
order_client_phone	Номер телефона	+12341231212
order_client_state	Штат	TX
order_client_zip	Индекс	12345
order_uid	Идентификатор заказа в системе клиента	12345-NB
order_custom_numeric	Числовое пользовательское поле	123456789
order_custom_text	Текстовое пользовательское поле	12345-NB-123
order_is_activation	Флаг активации/привязки	order_is_activation
project	Название проекта	promosite

Поля ответа:

Имя	Описание	Пример
amount	Сумма заказа	100.99
client_card_number	Маскированный номер карты	4111****1111
client_card_type	Тип карты*	mastercard
client_address	Адрес	123 Main Street
client_city	Город	Springfield
client_country	Страна	US
client_email	E-mail	mail@example.com
client_ipaddr	IP-адрес	123.123.123.123
client_cardholder	Имя держателя карты	Ivan Ivanov

client_phone	Номер телефона	12341231212
client_state	Штат	TX
client_postal_code	Индекс	12345
created	Дата создания заказа	2022-01-01 12:12:12
lastoperationdate	Дата последней операции	2022-01-01 12:12:12
order_id	Уникальный идентификатор заказа	689b987e82aa17d515180778 1360baf9
merchant_order_id	Идентификатор заказа в системе клиента	12345-NB
status	Статус	sale
issuer_id	Код банка-эмитента	123456
issuer_country	Страна банка-эмитента	US
issuer_name	Название банка-эмитента	Bank
custom_numeric	Числовое пользовательское поле	123456789
custom_text	Текстовое пользовательское поле	12345-NB-123
secure3d	Признак 3-D Secure	1
secure3d_mod	Режим работы 3-D Secure	attempt

* - возможные значения типа карты:

Константа	Тип
mir	МИР
visa	Visa
mastercard	MasterCard
amex	American Express
discover	Discover
cup	UnionPay

4.2.3.2. Список операций (GET /operation/)

Принимаемые параметры:

Имя	Пример
operation_created_from	2011-05-01
operation_created_to	2011-05-31
operation_message	Success
operation_status	success

operation_type	authorize
order_is_activation	Флаг активации/привязки
project	Название проекта
operation_type!=	prepare3d

Поля ответа:

Имя	Описание	Пример
amount	Сумма операции	100.99
created	Дата проведения операции	2011-05-31 12:13:15
order_id	ID ордера	689b987e82aa17d5151807781360baf9
merchant_order_id	Идентификатор заказа в системе клиента	12345-NB
status	Статус операции	success
type	Тип операции	authorize
authcode	Код авторизации	12345
responsecode	Код ответа	80
message	Описание результата	Success

4.2.3.3. Статистика по кодам ответов (GET /statbymessage/)

Принимаемые параметры:

Имя	Описание	Пример
operation_created_from	Дата создания операции (от)	2011-05-01
operation_created_to	Дата создания операции (до)	2011-05-31

Поля ответа:

Имя	Описание	Пример
message	Код ответа	Success
count	Количество	10345
percent	Доля среди всех ответов в процентах	15

4.2.3.4. Информация об ордере (GET /order/:id)

Дополнительные параметры не принимаются.

Поля ответа повторяют набор полей из раздела “Список ордеров”, дополнительно передаются следующие поля:

Имя	Описание	Пример
message	Пояснение к статусу	Declined by site rule
description	Описание платежа	Электронный товар (заказ #123456789)
descriptor	Дескриптор терминала	TRMDSCR12
issuer_name	Название банка-эмитента	Sberbank
issuer_country	Код страны банка-эмитента	RUS

Также дополнительно передается элемент operations, каждый элемент которого содержит поля из раздела “Список операций”.

Если заказ оплачивается через GDS, дополнительные параметры состояния ордера передаются в элементе GDS.

4.2.3.5. Отмена авторизации, возвраты Cancel (POST /order/:id/cancel)

Тип операции в ответе зависит от состояния ордера:

Статус ордера	Тип операции
authorize	reversal
sale	refund

Принимаемые параметры:

Имя	Описание	Пример
amount *	Сумма операции	100.99
extended *	Дополнительные параметры	См. 11. Формирование параметра extended

* - опциональное поле

Поле amount позволяет провести частичную отмену авторизации (authorize) или частичный возврат (refund) на указанное значение параметра amount в соответствии с тем, в каком статусе находится ордер. Передаваемая сумма должна быть не больше суммы, которая была фактически проведена операцией authorize или списана операцией settle. Если ранее уже проводились частичные рефанды, сумма не должна превышать остаточную сумму, которая не была возвращена.

ВНИМАНИЕ! Частичная отмена авторизации возможна только при условии эквайринга банковских карт через банк ПАО “ВТБ” и проводится только однократно.

Если сумма не передается, операция проводится на полную сумму, либо на остаточную сумму, если ранее уже проводились частичные операции reversal или refund.

Допускается проведение нескольких операций частичного refund, до тех пор, пока сумма списанных средств не будет возвращена полностью.

Поля ответа:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d5151807781360baf9
merchant_order_id	Идентификатор заказа в системе клиента	123456789
message	Описание результата	Success
status	Статус операции	success
time	Дата выполнения запроса	2011-05-31 12:13:15

4.2.3.6. Списание денежных средств Settle (POST /order/:id/settle)

Принимаемые параметры:

Имя	Описание	Пример
amount *	Сумма операции	100.99
longrecord *	«Длинная запись»	01 MOW LEDBTY LEDBTY MOW....
extended **	Дополнительные параметры	См. 11. Формирование параметра extended

* - Поле amount позволяет провести частичный settle. Передаваемая сумма не должна превышать авторизованную сумму. Операция выполняется только один раз, независимо от того, какая сумма была указана. Если поле amount не передается, будет проведена операция settle на всю авторизованную сумму.

** - Опциональное поле

Поля ответа:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d5151807781360baf9
merchant_order_id	Идентификатор заказа в системе клиента	123456789
message	Описание результата	Success
status	Статус операции	success
time	Дата выполнения запроса	2011-05-31 12:13:15

4.2.3.7. Применение Rebill (POST /order/:id/rebill)

Принимаемые параметры:

Имя	Описание	Пример
merchant_order_id	Идентификатор заказа в системе клиента	123456789
amount	Сумма операции	100.99
cvn *	Код CVV/CVC	123
ip	IP-адрес	123.123.123.123
description *	Описание платежа	Электронный товар (заказ #123456789)
project *	Название проекта	promosite
custom_numeric *	Числовое пользовательское поле (8 байт)	123456789
custom_text *	Текстовое пользовательское поле (до 12 символов)	12345-NB-123
terminal *	Дескриптор терминала	MYSHOP.RU
longrecord *	«Длинная запись»	01 MOW LEDBTY LEDBTY MOW.
extended *	Дополнительные параметры	См. 11. Формирование параметра extended

* – Опциональное поле

Передача параметра CVN возможна только после подтверждения соответствия PCI DSS на данные операции с ПДК.

Поля ответа:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d5151807781360baf9
merchant_order_id	Идентификатор заказа в системе клиента	123456789
status	Статус операции	success
message	Описание результата	Success
descriptor	Дескриптор терминала	TRMDSCR12
authcode	Код авторизации	12345
time	Дата выполнения запроса	2011-05-31 12:13:15

4.2.3.8. Применение Rebill3d (POST /order/:id/rebill3d)

Набор полей в ответе соответствует операции rebill. Набор параметров запроса также соответствует операции rebill, но дополнительно поддерживается еще два опциональных параметра:

Имя	Описание	Пример
return_success_url	URL для перенаправления клиента в случае успешной операции должен быть передан согласно стандарту RFC 3986	http://example.com/success?
return_failure_url	URL для перенаправления клиента в случае ошибки должен быть передан согласно стандарту RFC 3986	http://example.com/failure?

Параметры return_success_url и return_failure_url используются для перенаправления клиента при завершении процедуры 3D Secure и имеют приоритет над соответствующими параметрами, установленными в свойствах сайта.

Если карта поддерживает 3D Secure, в ответе устанавливается тип операции prepare3d, противном случае - rebill.

4.2.3.9. Оплата через Google Pay (POST /order/googlepay)

Использование сервиса Google Pay на сайте продавца

Схема взаимодействия *Продавец - GooglePay - GateLine*.

1. Пользователь нажимает кнопку GooglePay на сайте магазина.
2. Сайт магазина пересылает результат получения платежных данных на сервер магазина.
3. Сервер магазина передает полученные платежные данные и другие параметры в GateLine через вызов метода API `/order/googlepay``.
4. GateLine выполняет расшифровку платежных данных и взаимодействует с банком для выполнения оплаты с выполнением стандартных процедур. Для нетокенизированного метода оплаты потребуется прохождение клиентом 3D авторизации.
5. Продавец получает результат оплаты.

Принимаемые параметры: * – Опциональное поле

Имя	Описание	Пример
merchant_order_id	Идентификатор заказа в системе клиента	123456789
amount	Сумма операции	100.99
payload**	paymentMethodData.tokenizationData.token, полученный от google pay. При передаче необходимо закодировать в base64	WdodCI6IjYyNSIsIldpbmRvd1dpZHRolj oiMTM0MyJ9fQ==

extended.client_browser_info**	Блок данных о браузере клиента используется в терминалах банка ПАО ВТБ.	См. Формирование параметра extended
extended.cardnetwork**	PaymentMethodData.info.cardNetwork, полученный от google pay	См. Формирование параметра extended
country *	Страна (ISO 3166-1 alpha-2)	US
state *	Штат (ISO 3166-2)	TX
city *	Город	Springfield
postal_code *	Индекс	12345
address *	Адрес	123 Main Street
phone *	Номер телефона (до 20 символов)	+12341231212
email *	E-mail	mail@example.com
ip	IP-адрес клиента	123.123.123.123
description *	Описание платежа	Электронный товар (заказ #123456789)
return_success_url *	URL для перенаправления клиента в случае успешной операции должен быть передан согласно стандарту RFC 3986	http://example.com/success?
return_failure_url *	URL для перенаправления клиента в случае ошибки должен быть передан согласно стандарту RFC 3986	http://example.com/failure?
project *	Название проекта	promosite
terminal *	Дескриптор терминала	MYSHOP.RU
timeout *	Задержка отправки ответа, значение указывается в секундах.	60
longrecord *	«Длинная запись»	01 MOW LEDBTY LEDBTY MOW....
extended *	Дополнительные параметры	См. Формирование параметра extended

** – при использовании скрипта, предоставленного сотрудниками тех. поддержки, параметры payload и client_browser_info формируются автоматически в искомой кодировке

Поля ответа:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d5151807781360baf9
merchant_order_id	Идентификатор заказа в системе клиента	123456789
status	Статус операции	success
message	Описание результата	Success
descriptor	Дескриптор терминала	TRMDSCR12
authcode	Код авторизации	12345
responsecode	Двухсимвольный код ответа от банка	00
time	Дата выполнения запроса	2011-05-31 2:13:15

4.2.3.10. Оплата через Yandex Pay (POST /order/yapay)

Использование сервиса Yandex Pay на сайте продавца

Схема взаимодействия *Продавец - YandexPay - GateLine.*

1. Пользователь нажимает кнопку YandexPay на сайте магазина.
2. Сайт магазина пересылает результат получения платежных данных на сервер магазина.
3. Сервер магазина передает полученные платежные данные и другие параметры в GateLine через вызов метода API `/order/yapay``.
4. GateLine выполняет расшифровку платежных данных и взаимодействует с банком для выполнения оплаты с выполнением стандартных процедур. Для нетокенизированного метода оплаты потребуется прохождение клиентом 3D авторизации.
5. Продавец получает результат оплаты.

Принимаемые параметры:

Имя	Описание	Пример
merchant_order_id	Идентификатор заказа в системе клиента	123456789
amount	Сумма операции	100.99
payload**	paymentMethodData.tokenizat ionData.token, полученный от yandex pay. При передаче необходимо закодировать в base64	WdodCl6IjYyNSIsIldpbmRvd1dpZHRoIj oiMTM0MyJ9fQ==

extended.client_browser_info**	Блок данных о браузере клиента используется в терминалах банка ПАО ВТБ.	См. Формирование параметра extended
extended.cardnetwork**	PaymentMethodData.info.card Network, полученный от google pay	См. Формирование параметра extended
country *	Страна (ISO 3166-1 alpha-2)	US
state *	Штат (ISO 3166-2)	TX
city *	Город	Springfield
postal_code *	Индекс	12345
address *	Адрес	123 Main Street
phone *	Номер телефона (до 20 символов)	+12341231212
email *	E-mail	mail@example.com
ip	IP-адрес клиента	123.123.123.123
description *	Описание платежа	Электронный товар (заказ #123456789)
return_success_url *	URL для перенаправления клиента в случае успешной операции должен быть передан согласно стандарту RFC 3986	http://example.com/success?
return_failure_url *	URL для перенаправления клиента в случае ошибки должен быть передан согласно стандарту RFC 3986	http://example.com/failure?
project *	Название проекта	promosite
terminal *	Дескриптор терминала	MYSHOP.RU
timeout *	Задержка отправки ответа, значение указывается в секундах.	60
longrecord *	«Длинная запись»	01 MOW LEDBTY LEDBTY MOW....
extended *	Дополнительные параметры	См. Формирование параметра extended

* – Опциональное поле

** – при использовании скрипта, предоставленного сотрудниками тех. поддержки, параметры payload и client_browser_info формируются автоматически в искомой кодировке

Поля ответа:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d5151807781360baf9
merchant_order_id	Идентификатор заказа в системе клиента	123456789
status	Статус операции	success
message	Описание результата	Success
descriptor	Дескриптор терминала	TRMDSCR12
authcode	Код авторизации	12345
responsecode	Двухсимвольный код ответа от банка	00
time	Дата выполнения запроса	2011-05-32 2:13:15

4.2.3.11. Оплата через Apple Pay (POST /order/applepay)

Схема взаимодействия *Продавец - ApplePay - GateLine*.

1. Покупатель нажимает кнопку Apple Pay на сайте магазина, платежной форме GateLine
2. По нажатию на кнопку веб-сайт получает от Apple Url запроса сессии, который необходимо передать в ПШ GateLine. Полученный ответ необходимо вернуть на сайт.
3. Покупатель в браузере или в мобильном приложении видит диалог с привязанной к сервису Apple Pay картой, суммой и наименованием магазина в сторону которого производится оплата.
4. Покупатель инициирует оплату через отпечаток пальца, сканирование лица или ввод пароля. В ответ от Apple возвращается платежный токен в зашифрованном виде
5. Веб-сайт или приложение отправляет токен оплаты вместе со сведениями о покупке в свой сервер и отправляет запрос в GateLine.
6. Для выполнения платежа серверная часть GateLine обрабатывает покупку и отправляет платежные данные в банк-эквайер.
7. По результату оплаты производится редирект клиента на указанный redirect_url
8. Покупатель видит результат оплаты на странице сайта или в приложении.

*Для сценария, в котором кнопка Apple Pay расположена на платежной форме GateLine, каких-то действий со стороны мерчанта не требуется.

Метод Apple Pay (POST /checkout/applesession)

Принимаемые параметры:

Имя	Описание	Пример
validation_url	url запроса сессии ApplePay	https://apple-paygateway.apple.com/paymentservices/startSession

Поля ответа:

Имя	Описание	Пример
apple_session	Json с параметрами, который необходимо вернуть на фронт	{"epochTimestamp":1623937803510,"expiresAt":1623941403510,"**","retries": 0}

Метод Apple Pay (POST /order/applepay)

Имя	Описание	Пример
merchant_order_id	Идентификатор заказа в системе клиента	123456789
amount	Сумма операции	100.99
payload	paymentMethodData.tokenizationData.token, полученный от Apple Pay. При передаче необходимо закодировать в base64	WdodCl6IjYyNSIsIldpbmRvd1d pZHRoljoiMTM0M0MyJ9fQ==
country *	Страна (ISO 3166-1 alpha-2)	US
state *	Штат (ISO 3166-2)	TX
city *	Город	Springfield
postal_code *	Индекс	12345
address *	Адрес	123 Main Street
phone *	Номер телефона (до 20 символов)	+12341231212
email *	E-mail	mail@example.com
ip	IP-адрес клиента	123.123.123.123
description *	Описание платежа	Электронный товар (заказ #123456789)

return_success_url *	URL для перенаправления клиента в случае успешной операции должен быть передан согласно стандарту RFC 3986	http://example.com/success?
return_failure_url *	URL для перенаправления клиента в случае ошибки должен быть передан согласно стандарту RFC 3986	http://example.com/failure?
project *	Название проекта	promosite
terminal *	Дескриптор терминала	MYSHOP.RU
timeout *	Задержка отправки ответа, значение указывается в секундах.	60
longrecord *	«Длинная запись»	01 MOW LEDBTY LEDBTY MOW....
extended *	Дополнительные параметры	См. Формирование параметра extended

Принимаемые параметры: * – Опциональное поле

Поля ответа:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d515180778 1360baf9
merchant_order_id	Идентификатор заказа в системе клиента	123456789
status	Статус операции	success
message	Описание результата	Success
descriptor	Дескриптор терминала	TRMDSCR12
authcode	Код авторизации	12345
responsecode	Двухсимвольный код ответа от банка	00
time	Дата выполнения запроса	2011-05-33 2:13:15

4.2.3.12. Оплата через SberPay

Подключение SberPay на платежной форме GateLine

В данном варианте предполагается размещение на платежной форме GateLine в качестве одного из методов оплаты кнопки SberPay и редирект плательщика в канал Web или mWeb на данную платежную форму.

Подключение SberPay на платежной форме GateLine осуществляется на стороне GateLine на этапе настройки.

Запрос на платежную форму GateLine отправляется методом POST /checkout/pay в соответствии с пунктом 4.4.1. "Инструкции по интеграции Gateline". Ответ на запрос будет содержать на в поле "redirect" ссылку на платежную форму. На платежной форме будет в том числе кнопка SberPay.

Пример ответа на запрос **POST /checkout/pay:**

```
<?xml version="1.0" encoding="utf-8"?>
<checkout>
  <pay>
    <redirect>https://checkout.gateline.ru/pay?token=eaae879f561bdf9dd4facd08d12225d</redirect>
  </pay>
</checkout>
```

Подключение SberPay в мобильном приложении

В мобильном приложении торгово-сервисного предприятия кроме варианта, приведенного в предыдущем разделе, можно организовать оплату по SberPay без перехода на платежную форму GateLine - через deer_link в приложение Сбербанк Онлайн.

Для получения deer_link в приложение Сбербанк Онлайн нужно сделать запрос в платежный шлюз GateLine POST /checkout/pay с дополнительным параметром paysystem=sberpay

В этом случае, помимо ссылки на платежную форму в "redirect" также вернется дополнительный параметр "deer_link" со ссылкой на переход в приложение Сбербанк Онлайн.

Пример ответа на запрос **POST /checkout/pay + paysystem=sberpay:**

```
<?xml version="1.0" encoding="utf-8"?>
<checkout>
  <pay>
    <redirect>https://checkout.sandbox.gateline.net:18310/pay?token=2f4012ae7d31163bbd6920cc03ab5c56
    </redirect>
    <sberpay>sberpay://invoicing/v2?bankInvoiceId=e165c1f2cd604f77aee1c27ad4c0eb93&operationType=Web2App
    </sberpay>
  </pay>
</checkout>
```

При интеграции сценария mweb2app на устройствах apple необходимо реализовывать процедуру перебора диплинков. Это связано с различными версиями приложений, установленных на устройствах пользователей. Подробная инструкция по реализации процедуры доступна в [документе Сбера](#).

Уведомления об оплате

При оплате методом SberPay в уведомлении об оплате передается поле с указанием платежной системы payment_type=sberpay

Пример уведомления при оплате **SberPay:**

```
amount=332.00&authcode=777777&check-
sum=d023a50edefa19448f4deed1982d28b60af4f598&descriptor=SberPayTERMINAL&email=mobde-
```

velop%40mail.ru&masked_pan=3c49a8ba3e8fd61d81f17723a66531f7c90227d9339909330f6e478c7e6221cf&merchant_order_id=13151&message=Success&operation=authorize&order_id=c89611dabae1ee11a19d1d6a29244cb9&payment_type=sberpay&provider=card&responsecode=00&status=success&time=2024-03-14%2007%3A25%3A13

4.2.3.13. Тестовый запрос (GET /test/ping)

Параметры не принимаются.

Ответ содержит единственный элемент:

Имя	Описание	Пример
time	Текущее время	2011-05-31 12:13:15
timeout	Задержка отправки ответа, значение указывается в секундах.	60

4.3.Direct API. Прямой процессинг

Для ТСП, планирующий интеграцию с использованием PCI DSS необходимо воспользоваться двумя методами «authorize или authorize3d». Все запросы, отправляемые на хост Direct API необходимо подписывать сертификатом, который выдается службой технической поддержки ПШ после согласования метода интеграции.

4.3.1. Применение Authorize (POST /order/authorize)

Принимаемые параметры:

Имя	Описание	Пример
merchant_order_id	Идентификатор заказа в системе клиента	123456789
amount	Сумма операции	100.99
card_number	Номер карты	4111111111111111
expiration_month	Дата истечения (месяц)	04
expiration_year	Дата истечения (год)	2016
cvn	Код CVV/CVC	123
cardholder	Держатель карты	Gill Bates
country *	Страна (ISO 3166-1 alpha-2)	US
Имя	Описание	Пример
state *	Штат (ISO 3166-2)	TX
city *	Город	Springfield

postal_code *	Индекс	12345
address *	Адрес	123 Main Street
currency *	Цифровой код валюты согласно ISO 4217	643
exchange_rate *	Курс валюты относительно рубля	0.05
currency_amount *	Сумма заказа отображаемая на платежной форме в случаи если currency не равно 643	123
email *	E-mail	mail@example.com
ip	IP-адрес клиента	123.123.123.123
description *	Описание платежа	Электронный товар (заказ #123456789)
return_success_url *	URL для перенаправления клиента в случае успешной операции должен быть передан согласно стандарту RFC 3986	http://example.ru/success?
return_failure_url *	URL для перенаправления клиента в случае ошибки должен быть передан согласно стандарту RFC 3986	http://example.ru/failure?
project *	Название проекта	promosite
terminal *	Дескриптор терминала	MYSHOP.RU
timeout *	Задержка отправки ответа, значение указывается в секундах.	60
longrecord *	«Длинная запись»	01 MOW LEDBTY LEDBTY MOW....
extended *	Дополнительные параметры	См. 11. Формирование параметра extended

* – Опциональное поле

Поля ответа:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d5151807781360baf9

merchant_order_id	Идентификатор заказа в системе клиента	123456789
status	Статус операции	success
message	Описание результата	Success
descriptor	Дескриптор терминала	TRMDSCR12
authcode	Код авторизации	12345
payment_type	Тип провайдера	card
responsecode	Двухсимвольный код ответа от банка	00
time	Дата выполнения запроса	2011-05-31 12:13:15

Возможно использовать только после подтверждения соответствия PCI DSS на данные операции с ДДК и ПДК.

Пример успешного ответа:

```
<operation>
  <ТИП_ОПЕРАЦИИ>
    <message>Success</message>
    <status>success</status>
    <time>2011-05-01 15:15:15</time>
    ... прочие поля ответа ...
  </ТИП_ОПЕРАЦИИ>
</operation>
```

Пример неудачного ответа:

```
<operation>
  <ТИП_ОПЕРАЦИИ>
    <message>Decline</message>
    <status>failed</status>
    <time>2011-05-01 15:15:15</time>
    ... прочие поля ответа ...
  </ТИП_ОПЕРАЦИИ>
</operation>
```

4.3.2. Применение Authorize3d (POST /order/authorize3d)

Запрос order/Authorize3d передается в случае необходимости использования 3ds. Набор параметров запроса соответствует операции authorize.

Если карта поддерживает 3D Secure, в ответе устанавливается тип операции prepare3d, противном случае - authorize.

Поля ответа:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d51518077813 60baf9
threeDSMethodURL*	url ссылка, которую необходимо открыть в скрытом iframe в браузере клиента. опциональное поле (актуально для эквайера ВТБ)	https://3dsp2.multicarta.ru/device-information
threeDSServerTransID*	Идентификатор транзакции, выданный 3DS Сервером, должен быть использован при формировании ThreeDSMethodData. опциональное поле (актуально для эквайера ВТБ)	da12fff4-b196-4c28-8839ee06bcd53838
form*	Ссылка на ацс форму, на которую необходимо средиректить клиента. опциональное поле	action=https%3A%2F%2F3dsp2.multicarta.ru%3A443%2F%3FCSMName%3DFIID%2FVB24&creq=ewogICJhY3NUcmFuc0IEliA6IClyYTdiMzkyMi02MzQ0LTQ3NDgtYWQxMy1hZmU4MTY0ZjY2ZjMiLAogICJ0aHJIZURTU2VydmVyVHJhbnNJRCIgOiAiZGExMmZmZjQtYjE5Ni00YzI4LTg4MzktZWUwNmJjZDUzODM4IiwKICAiY2hnbGxlbmdIV2luZG93U2I6ZSIgOiAiMDMiLAogICJtZXNzYWdlVHlwZSIgOiAiQ1JlcSIsCiAgIm1lc3NhZ2VWZXJzaW9uIiwKIClyLjEuMCIkFkQ%3D%3D
status	Статус операции	success
message	Описание результата	Success
provider	Тип провайдера	card
descriptor	Дескриптор терминала	TRMDSCR12
time	Дата выполнения запроса	2011-05-31 12:13:15

* – Опциональное поле

Описание блока client_browser_info

Блок client browser info обязателен в случае, если используется терминал в банке ПАО ВТБ. Блок необходимо передавать внутри параметра extended.

Блок client_browser_info имеет json структуру и содержит следующие параметры:

Имя	Обязательность	Описание
browserAcceptHeader	Да	Содержимое HTTP-заголовков, отправленных из браузера клиента. Максимальное значение – 2048 символов
browserColorDepth	Да	Значение, представляющее битовую глубину цветовой палитры для отображения изображений, в битах на пиксель. Максимальное значение – 2 символа. Возможные значения: 1 - 1 бит; 4 - 4 бита; 8 - 8 битов; 15 - 15 битов; 16 - 16 битов; 24 - 24 бита; 32 - 32 бита; 48 - 48 бито
browserIP	Да	IP-адрес браузера. Возможные форматы значения: IPv4-адрес указан в виде четырех групп чисел в десятичной системе счисления, разделенных символом «.». Например, 1.12.123.255. IPv6-адрес указан в виде восьми групп чисел в шестнадцатеричной системе счисления, разделенных символом «:». Например, 2011: 0db8: 85a3: 0101: 0101: 8a2e: 0370: 7334
browserLanguage	Да	Язык браузера, указанный в IETF BCP47. Максимальное значение – 8 символов
browserScreenHeight	Да	Общая высота (в пикселях) экрана, отображаемого держателю карты. Максимальное значение – 6 символов

browserScreenWidth	Да	Общая ширина (в пикселях) экрана, отображаемого держателю карты. Максимальное значение – 6 символов
browserTZ	Да	Разница во времени между временем по UTC и местным временем браузера пользователя. Максимальное значение – 5 символов
browserUserAgent	Да	Содержимое HTTP-заголовка User-Agent. Максимальное значение – 2048 символов
deviceChannel	Да	Тип устройства, с которого инициирована транзакция. Возможные значения: 01 - мобильное приложение ТСП; 02 - браузер пользователя; 03 - 3DS Requestor
browserJavaEnabled	Да	Признак возможности выполнения JavaScript в браузере держателя карты. Возможные значения: true, false
WindowWidth	Да	Ширина окна браузера (в пикселях), в котором отображаются страницы сайта ТСП
WindowHeight	Да	Высота окна браузера (в пикселях), в котором отображаются страницы сайта ТСП

4.3.3. Применение result3dsmethod (POST /order/:id/result3dsmethod)

Запрос result3dsmethod дополнительный запрос, передается после ответа, полученного на запрос order/Authorize3d. Необходим для завершения дополнительной 3дс проверки, для получения Асс формы. Актуален только для эквайера ВТБ.

После получения threeDSServerTransID и threeDSMethodURL необходимо открыть скрытый фрейм в браузере клиента, с помощью которого данные о браузере передаются по протоколу HTTP на данный URL. Данный процесс согласно спецификации, EMV называется 3DS Method и возвращает в ThreeDSMethodNotificationURL параметр threeDSMethodData.

Ждать ответ необходимо в течении 10 секунд

Пример формирования Methoddata:

```
action=threeDSMethodURL
method=POST
threeDSMethodData = base64

{"threeDSServerTransID":"3ac7caa7-aa42-2663-791b-2ac05a542c4a","threeDSMethodNotificationURL":"урл, куда придет ответ"}
```

Поля запроса:

Имя	Описание	Пример
comp_ind	Значения Y (при получении ответа от threeDSMethodURL в течение 10 секунд) или N (при отсутствии ответа в течение 10 секунд)	Y

Поля ответа:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d5151807781360baf9
form	Ссылка на ацс форму, на которую необходимо средиректить клиента. опциональное поле (отсутствует если эквайер ВТБ)	
status	Статус операции	success
message	Описание результата	Success
provider	Тип провайдера	card
descriptor	Дескриптор терминала	TRMDSCR12
time	Дата выполнения запроса	2011-05-31 12:13:15

4.4.Платежный виджет

При проведении интеграции через API имеется возможность на сайте Интернет-магазина использования Платежного виджета GateLine - всплывающей формы для выбора инструмента оплаты и ввода реквизитов карты плательщика без визуального перехода на отдельную страницу Платежного шлюза.

В данном варианте фактически ввод реквизитов карты происходит на стороне Платежного шлюза, а визуально - на стороне сайта Интернет-магазина. Отсутствие перенаправления на другие страницы положительно воспринимается пользователями и повышает конверсию платежей.

Цель виджета - упростить интеграцию Организации с Платежным шлюзом GateLine. Для интеграции достаточно подключить на страницу Интернет-магазина специальный скрипт для открытия платежной формы в виде всплывающего окна iframe на текущей странице сайта поверх неё.

Для установки Платежного виджета на необходимо:

Прописать на сайте Интернет-магазина в разделе head следующий скрипт:

```
//При работе с тестовой средой Платежного шлюза GateLine
<script
src="https://checkout.sandbox.gateline.net/js/gateline/widget.js"></script>
//При работе с промышленной средой Платежного шлюза GateLine
<script src="https://checkout.gateline.ru/js/gateline/widget.js"></script>
```

Создать объект Widget:

```
let widget = new GL.Widget({style: { width: 123, height: 123}});
```

Параметры стиля конструктора - не обязательные. Если не назначать ширину и высоту - виджет подстроится под размер экрана.

Далее есть два варианта инициирования появления платежной формы:

3.1. Первый вариант - через метод `widget.checkout_pay` с передачей параметров, соответствующих параметрам одноименного метода API GateLine `checkout/pay` (согласно п.п. 4.4.1. настоящей Инструкции). При этом обязательно надо передавать идентификатор сайта и его пароль, они используются при формировании чек-суммы.

Пример:

```
widget.checkout_pay({      site: "site_id",           // идентификатор
магазин в gateline      api_key: "1231",           // ключ магазина в
gateline      amount: 12.34,           // сумма оплаты      description:
"Назначение платежа", // назначение платежа
      merchant_order_id: "123456",           // id заказа в Интернет-магазине
    },
  // Функции, вызываемые после получения результата оплаты. Аргумент result будет
содержать параметрам редиректа на результат оплаты      on_result:
function(result) { ... }, //Вызывается как только виджет получает от
api.gateline ответ с результатом транзакции.
      on_success: function(result) { ... }, //ДЕЙСТВИЕ при успешной оплате
on_fail: function(result) { ... }, //ДЕЙСТВИЕ при неуспешной оплате
    }
  });
```

3.2. Другой вариант - это с помощью метода `checkout_pay({token: token})`. В этом варианте необходимо через метод API GateLine `checkout/pay` (согласно п.п. 4.2.1. настоящей Инструкции) получить параметр `token` из ранее сформированной ссылки на платежную форму и передать его в виджет.

Пример:

```
checkout_pay({site: site, api_key: api_key, merchant_order_id: ...});
```

Параметры уведомления об оплате соответствуют параметрам URL редиректа на страницу результата.

```
{  
  
  status: "success" | "failed", order_id:  
  "order_id gateline", merchant_order_id:  
  "order id магазина", message: "Описание  
  результата операции", code: "код  
  ответа", checksum: "чек-сумма",  
  
}
```

После оплаты виджет отобразит окно с предложением вернуться в магазин.

```
widget.checkout_activation({...});
```

5. Уведомления

5.1. Общая информация

Уведомления предназначены для асинхронной передачи информации от системы магазину.

По умолчанию доставляются после каждой успешной операции, проведенной через интерфейс партнера или через API. Исключением являются операции, связанные с активацией (authorize и reversal), для них уведомления не доставляются.

При необходимости для магазина может быть применена собственная политика доставки уведомлений.

5.2. Требования к сайту, принимающему уведомления

Уведомления отправляются по протоколу HTTPS в одном из форматов по выбору:

- XML (Content-Type: text/xml);
- HTML (Content-Type: application/x-www-form-urlencoded).

На сайте партнера на указанном URL должна быть настроена базовая HTTP-аутентификация с реквизитами доступа, соответствующими тем, что указаны в свойствах сайта. Если аутентификация не настроена, уведомление не доставляется.

Перед отправкой уведомления, отправляется запрос методом GET для проверки наличия базовой HTTP-аутентификации, после отправляется запрос отправляется методом POST на URL, указанный в свойстве сайта «URL для доставки уведомлений».

5.3. Контроль доставки уведомления

Если сайт принял и обработал уведомление он должен сформировать ответ с HTTP-статусом 200, тело ответа должно состоять из слова «SUCCESS» латиницей в верхнем регистре. Окружающие ответ пробельные символы игнорируются. В этом случае система будет считать, что уведомление доставлено и обработано.

Если первое уведомление не было доставлено, система производит еще 3 попытки через

определенные промежутки времени. Если все 4 попытки доставить уведомление оказались неудачными, система не предпринимает никаких дополнительных действий. Список ошибок доставки уведомлений можно посмотреть в разделе «Журналы -> Ошибки уведомлений».

5.4. Формат уведомления

Формат сообщения соответствует стандартному формату ответа системы. Набор полей зависит от типа операции, на которую передается уведомления и идентичен набору полей, который передается в синхронном ответе на операцию в API

5.5. Верификация настроек уведомлений

Уведомления доставляются только после подтверждения ссылок на стороне тех. поддержки шлюза. После каждого изменения настроек уведомлений требуется повторное подтверждение.

Для продуктивной среды необходимо заранее сообщать URL на который планируется получать уведомления, т.к. адреса добавляются в списки разрешенных на стороне шлюза Gateline.

Тестовое уведомление (xml):

```
<?xml version="1.0" encoding="utf-8"?>
<operation>
  <test>
    <message>test</message>
  </test>
</operation>
```

Тестовое уведомление (urlencoded):

operation=test&message=test

Операции, на которые доставляются уведомления: confirmation, authorize, reversal, settle, refund:

Поля ответа:

Имя	Описание	Пример
merchant_order_id	Идентификатор заказа в системе клиента	123456789
message	Описание результата	Success
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d5151807781360baf9
provider	Тип провайдера	card
status	Статус операции	success
time	Дата выполнения запроса	2011-12-12 12:12:12
amount*	Сумма <i>Используется в confirmation</i>	123
description*	Описание заказа <i>Используется в confirmation</i>	Товар (#123456789)
email	eMail введенный плательщиком на платежной форме	test@test.ru
payment_type	Метод оплаты принимает значение: card – оплата выполнена с использованием банковской карты; sbr – оплаты выполнена через СБП	card
saved_card*	Признак сохранения карты на Платежной формы	1
responsecode*	Код ответа <i>Используется в authorize</i>	000

* – Опциональное поле

confirmation

```
<operation>
<confirmation>
<amount>500.00</amount>
<description>TEST-PAY</description>
<merchant_order_id>12345</merchant_order_id>
<provider>card</provider>
</confirmation>
</operation>
```

authorize

```
<operation>
<authorize>
<authcode>XXX09X</authcode>
<descriptor>TEST-TERM</descriptor>
<merchant_order_id>12345</merchant_order_id>
<email>test@test.ru</email>
<message>Success</message>
<order_id> rfm50cd3xqa419jp2kn0u2ehm0bZR9rd1</order_id>
<provider>card</provider>
<responsecode>000</responsecode>
<status>success</status>
<time>2000-01-01 00:00:01</time>
</authorize>
</operation>
```

settle

```
<operation>
<settle>
<merchant_order_id>12345</merchant_order_id>
<message>Success</message>
<order_id>rfm50cd3xqa419jp2kn0u2ehm0bZR9rd1</order_id>
<provider>card</provider>
<status>success</status>
<time>2000-01-01 00:00:03</time>
</settle>
</operation>
```

5.6. Активация

По умолчанию при проведении активации система не доставляет уведомление на операцию authorize, которой блокировалась случайная сумма. Через некоторое время система отменяет блокировку, при этом уведомление на операцию reversal также не доставляется.

Уведомление для успешной активации передается в стандартном формате уведомления, при этом типом операции является activation, передаются следующие поля:

Имя	Описание	Пример
order_id	Уникальный идентификатор заказа в системе	689b987e82aa17d5151807781360
merchant_order_id	Идентификатор заказа в системе клиента	123456789
status	Статус операции	success
secure3d	Передается значение 1, если была произведена авторизация с использованием 3D Secure.	1
message	Описание результата	Activation success
time	Дата завершения операции	2011-05-31 12:13:15

5.7. Подтверждение оплаты

Если в сайте установлено свойство “Запрашивать подтверждение операций”, перед проведением оплаты система будет отправлять уведомление и ожидать ответа.

Это уведомление должно быть обработано по общим правилам, и если необходимость такой операции подтверждается, должен быть сформирован ответ SUCCESS, в противном случае – любой другой.

Если был получен ответ, отличный от SUCCESS, или уведомление не удалось доставить по любой другой причине, операция отклоняется с сообщением “Confirmation failed”.

Уведомление на подтверждение оплаты доставляется однократно, повторных попыток доставки не производится. Если подтверждение не было получено после первой попытки, операция отклоняется. Механизм используется в случае, если актуальность заказа сохраняется ограниченный период времени (бронирование, заказ последней единицы товара на складе и т.п.).

Уведомление для подтверждения передается в стандартном формате уведомления, при этом типом операции является confirmation, передаются следующие поля:

Имя	Описание	Пример
merchant_order_id	Идентификатор заказа в системе клиента	123456789
amount	Сумма операции	100.99
description	Описание платежа	Электронный товар (заказ #123456789)

6. Обработка редиректов

6.1. Общая информация

После некоторых операций пользователь, в зависимости от статуса операции, перенаправляется на один из двух установленных партнером в настройках редиректов URL.

Если в сайте установлено свойство «Передавать параметры при редиректе», к этому адресу добавляется ряд параметров, которые указывают на результат проведения операции, статус, привязку к ордеру и т.д.

Если установленный в сайте URL уже содержит GET-параметры, они будут сохранены при редиректе. Если имя одного из этих параметров совпадает с тем, которое устанавливает система при редиректе, будет возвращен только системный параметр.

URL может быть установлен двумя способами:

- Свойства сайта «URL возврата при успешной операции» и «URL возврата при ошибке». Используются по умолчанию.
- Параметры запроса в API `return_success_url` и `return_failure_url`. Эти параметры являются опциональными, и могут перекрывать значения, установленные в сайте.

Пользователь перенаправляется на сайт клиента в двух случаях:

1. Завершение авторизации через 3D Secure.
2. Завершение процедуры оплаты или активации через платежную форму.

6.2. Проверка контрольной суммы

Контрольная сумма передается как параметр `checksum`.

Контрольная сумма вычисляется как HMAC-SHA1-сумма строки, составленной из пар «имя=значение», отсортированных по имени в алфавитном порядке. Пары разделяются символом «;» (точка с запятой), в качестве пароля для вычисления контрольной суммы используется пароль сайта.

Параметры, которые присутствовали в изначальном URL, обрабатываются при вычислении checksum на общих основаниях.

6.3. Список передаваемых параметров

Название	Описание	Пример
<code>message</code>	Описание результата	Success
<code>status</code>	Статус операции	Success
<code>order_id *</code>	ID ордера	689b987e82aa17d5151807781360ba f9
<code>merchant_order_id *</code>	Идентификатор заказа в системе клиента	12345abc
<code>code **</code>	Код ошибки	504

* - поле может не передаваться, если установлен статус error

** - поле передается только для статуса error

6.4.Расшифровка статуса

Поле status может принимать следующие значения:

Значение	Описание
success	Операция проведена успешно
failed	Операция была инициирована, но не завершилась удачно по какой-либо причине.
error	Возникла проблема, которая не позволяет запустить проведение операции.

Дополнительная информация о результате операции содержится в поле message.

7. Обработка результата процессинга

В случае работы через платежную или использования механизма 3D Secure система может сообщить магазину о результате выполнения операции асинхронно одним из следующих способов:

1. Параметры редиректа, которым система перенаправляет пользователя обратно в магазин.
2. Уведомление, которое доставляется для каждой успешной операции.

Несмотря на то, что параметры редиректа содержат всю необходимую информацию о результате проведения операции, использовать их рекомендуется только для отображения пользователю специфических страниц, например, сообщений об ошибках.

Уведомления доставляется более безопасным способом, чем параметры редиректа. Кроме этого, уведомление будет доставлено даже в том случае, если операция выполнялась успешно, а пользователь по какой-либо причине не дождался результата или не проследовал по перенаправлению.

8. Работа с 3D Secure

8.1. Общая информация

Если затребовано проведение авторизации с использованием 3D Secure система направляет номер карты клиента в MPI, чтоб узнать, участвует ли карта в программе 3D Secure. Возможны следующие ответы:

1. Карта не может быть авторизована с использованием 3D Secure. В этом случае система блокирует проведение авторизации и возвращает клиенту соответствующий ответ.
2. Карта может использовать 3D Secure, однако не участвует в программе (Non-Participation). Аутентификация клиента не требуется, и система проводит авторизацию. При этом в банк эквайер передается метка об использовании 3D Secure и на такую транзакцию распространяется перенос ответственности.
3. Карта участвует в программе 3D Secure. В этом случае проводится аутентификация: клиент направляется на сайт банка-эмитента, где вводит свой платежный пароль либо одноразовый пароль из SMS-сообщения.

Аутентификация клиента может завершиться одним из следующих результатов:

1. Клиент успешно прошел аутентификацию.
2. Аутентификация не проведена, но попытка проведения зафиксирована (Attempt).
3. Клиента не завершил аутентификацию либо произошла системная ошибка.

В первых двух случаях система проводит авторизацию и на транзакцию распространяется перенос ответственности. В последнем случае система отклоняет авторизацию.

Независимо от ответа MPI и результата аутентификации, система устанавливает в заказ флаг `secure3d=1`.

8.2. Прямой процессинг

Для того чтобы система попыталась провести авторизацию с использованием 3D Secure, вместо операций `authorize/rebill` необходимо запрашивать соответственно `authorize3d/rebill3d`.

Если было запрошено проведение операции `authorize3d/rebill3d`, возможны следующие ответы системы:

1. Карта поддерживает авторизацию через 3D Secure. В этом случае в ответе будет передана операция `prereq3d`, в поле `form` указаны параметры HTML-формы, которую необходимо отобразить пользователю.

Когда пользователь отправит форму (это может быть сделано автоматически), начнется процедура аутентификации на стороне банка-эмитента.

В случае успешного прохождения аутентификации, в системе проводится операция `authorize`, после чего пользователь перенаправляется на URL, указанный в свойстве сайта.

2. Аутентификация не требуется. В этом случае проводится авторизация с меткой `“attempt”`, в ответе будет передана операция `authorize`.

3. Авторизация с использованием 3D Secure невозможна. В ответ передается операция authorize со статусом failed.

8.3. Процессинг через платежную форму

При отправке запроса на оплату через платежную форму может быть передан флаг force3d, который указывает на то, что необходимо провести авторизацию с использованием 3D Secure. Это является эквивалентом проведения операции authorize3d при прямом процессинге.

8.4. Активация и 3D Secure

Для того, чтобы система попыталась провести авторизацию случайной суммы с использованием 3D Secure, в запросе activation необходимо передать параметр try3d.

Изначально система пытается провести авторизацию обычным способом. Если авторизация отклоняется, и у системы есть основания полагать, что по данной карте возможны платежи только через 3D Secure, автоматически следует вторая попытка провести авторизацию с 3D Secure.

Во втором случае в системе создаются два ордера с одинаковым значением merchant_order_id, из которых только второй может получить статус authorize (успешная авторизация).

Если авторизация была успешно проведена каким-либо образом, будет отправлено уведомление об успешной операции authorize.

8.5. Обработка формы 3D Secure

Если происходит прямой процессинг с использованием 3D Secure, в ответ на операцию authorize3d/rebill3d система возвращает параметр <form>, в котором находятся параметры HTML формы. Эту форму следует интерпретировать как стандартную строку в формате QUERYSTRING, алгоритм ее обработки следующий:

1. Разбить строку по символу "&", в результате образуется набор пар значений формата "name=value"
2. Каждая пара разбивается по символу "=", второе полученное значение следует пропустить через функцию URL Unescape (RFC 3986).
3. Сформировать HTML-форму, в ее атрибутах указать method="post", action="<action>", где <action> - соответствующий параметр, полученный из строки.
4. Остальные параметры оформляются как скрытые (type="hidden") с именами и значениями такими, какие были получены из строки.
5. Для удобства клиентов рекомендуется отправлять эту форму автоматически, т.к. она не содержит никаких данных, которые клиент может корректировать или вводить самостоятельно. Форма приведет клиента в интерфейс банка-эмитента, в котором осуществляется аутентификация.

9. Клиринг

9.1. Общая информация

Под клирингом подразумевается процедура списания заблокированной (авторизованной) суммы с карты клиента.

Система предоставляет партнерам возможность управлять клирингом самостоятельно на уровне каждого ордера. Если такой необходимости нет, существует возможность включить автоматический клиринг, при котором система будет проводить необходимые операции без вмешательства партнера.

Для выбора режима клиринга и изменения настроек сайта, связанных с ним (время задержки, разрешение операции ручного клиринга при работе в автоматическом режиме) необходимо обратиться в службу поддержки.

9.2. Автоматический режим

Если для магазина включен автоматический клиринг, система автоматически проводит операцию settle для каждого ордера.

Клиринг проводится через некоторое время после авторизации. Система позволяет установить время задержки для каждого сайта отдельно, по умолчанию оно составляет 6 часов.

Если блокировка была снята (операция reversal) до того, как был произведен клиринг, операция списания проведена не будет.

9.3. Ручной режим

Для того, чтобы провести клиринг выбранного ордера, нужно отправить в API запрос на операцию settle.

Операцию можно провести только один раз для каждого ордера. Ордер должен быть в статусе authorize.

Если для сайта включен автоматический клиринг, это не исключает возможности провести операцию settle в ручном режиме, если есть такая необходимость.

10. Проведение тестовых транзакций

10.1. Общая информация

Тестовые транзакции проводятся только через тестовый терминал. Доступ к тестовому терминалу можно получить через службу поддержки.

Тестовые карты:

Успешный платеж:	Не успешный платеж:
Номер карты: 4111 1111 1111 1111 Срок действия: 02/32 CVC: 123 Имя держателя карты: SUCCESSPAYMENT	Номер карты: 4111 1111 1111 1111 Срок действия: 09/29 CVC: 111 Имя держателя карты: ERRORPAYMENT

10.2. Использование 3D Secure

Тестовый терминал поддерживает возможность провести транзакцию с использованием 3D Secure. Указывая специальные значения в полях даты истечения карты (месяц и год), можно эмулировать обработку карт с разной степенью поддержки 3Ds. Эти значения можно как передавать в API запросах, так и указывать на платежной странице. Они будут обрабатываться специальным образом только в том случае, если запрошен процессинг с 3-D Secure (запрошена операция authorize3d или передан флаг force3d/try3d соответственно).

10.3. Роли пользователей

Главный аккаунт пользователя, который передается клиенту, является Администратором. Из этого аккаунта можно добавлять других пользователей, которые будут иметь доступ в интерфейс партнера. Для этих пользователей можно выбирать одну из ролей: Менеджер, Оператор, Сотрудник СБ.

В таблице перечислены роли и доступные им действия.

Роль	Доступные действия
Оператор	Просмотр: заказы и операции, сайты, проекты, журналы ошибок, аккаунт, статистика, аналитика
Администратор	Просмотр всех разделов, просмотр списка пользователей, добавление и редактирование пользователей
Менеджер	Права оператора, плюс проведение операций над заказами
Менеджер сайта	Доступ к просмотру и управлению заказами только одного сайта
Сотрудник СБ	Права оператора, плюс дополнительные: просмотр списка пользователей, добавление заказов в стоп-лист, удаление из стоп-листа

11. Формирование параметра extended

Параметр extended должен содержать JSON-структуру с дополнительной информацией о платеже:

Пример передачи объекта в параметре extended.extraobject:

```
{"extraobject":{"name1":"value1","name2":"value2",...}}
```

Пример передачи простого значения extended.extrascalar:

```
{"extrascalar":"value"}
```

Полный перечень возможных полей extended смотреть в “Примеры поля Extended”.

Параметр extended должен содержать JSON-структуру с дополнительной информацией о платеже:

Параметр	Описание	Параметр
extended.facilitator	Параметр для работы схем Фасилитаторской / DWO схем. Актуален только для работы в рамках эквайера ВТБ	{"facilitator_id": "12345"}
extended.electronic_receipt	Отправка чека в OrangeData для удовлетворения требований Ф3-54	{"electronic_receipt": { Структуру чека см. “Дополнения к API GateLine в рамках 54-ФЗ” }}
extended.electronic_receipt_mk	Отправка чека в МультиКарту для удовлетворения требований Ф3-54	{"electronic_receipt_mk": { "device": "auto", "client_id": "1", "password": 1, "request_id": "8", "document_type": 0, "lines": [{ "qty": 2500, "price": 10000, "pay_attribute": 4, "tax_id": 1, "description": "Булочка с маком" }, { "qty": 500, "price": 200000, "pay_attribute": 4, "tax_id": 2, "description": "Икра чёрная, баклажанная" }], "non_cash": [125000], "tax_mode": 1, "phone_or_email": "user@example.com", "place": "www.example.com", "full_response": false }}
extended.webrebill	Сохранение карт на платежной форме через привязку к идентификатору клиента	{"webrebill": {"allow_rebill": "1", "allow_save": "1", "customer_id": "123456"}}

extended.dsrp	<p>Для передачи данных по dsrp необходимо передать дополнительную информацию в extended. Необходимые параметры: - dsrp: флаг проведения транзакции по указанному протоколу;</p> <ul style="list-style-type: none"> - xid: ид транзакции, полученный от apple/android/etc, 40 символов; - cavv: криптограмма, полученная от apple/android/etc, 28 символов; - waid: идентификатор сервиса <p>для Apple Pay - «103»; для Android Pay - «216»; для Samsung Pay - «217» для Masterpass - masterpass_id</p>	{"cavv": "xxx", "dsrp": 1, "waid": 103, "xid": "xxx"}
extended.send_link	<p>Возможность передачи мерчантом флага на отправку sms/email клиенту ссылки на checkout</p> <p>Ссылка на платежную форму будет отправлена на полученные от мерчанта параметры phone / email</p>	{"send_link": {"email": "1", "sms": "1"}}

extended.mita	<p>Дополнительные атрибуты транзакции (Merchant Initiated Transaction attributes).</p> <p>Возможные значения: 1 – транзакция инициируется ТСП с вводом карточных данных из хранилища ТСП(Используется при рекуррентных платежах);</p> <p>2 – транзакция инициируется держателем карты с вводом карточных данных из хранилища ТСП(Используется при рекуррентных платежах);</p> <p>4 – флаг инитной транзакции.</p>	{"mita": "1"}
---------------	---	---------------

	Обязателен при последующем выполнении рекуррентного платежа	
extended.orig_order_id	<p>Идентификатор инитного платежа.</p> <p>Параметр используется при выполнении рекуррентных платежей, для связи рекуррентного платежа с установочным платежом.</p> <p>Источник данных - order_id поле ответа на инитную операцию. Обязателен при выполнении рекуррентного платежа</p>	{"orig_order_id": "689b987e82aa17d5151807781360baf9"}

extended.wallet	<p>Номер и наименование электронного кошелька получателя.</p> <p>Параметр следует передавать, если денежные средства, списываемые с карточного счёта держателя, предназначены для пополнения баланса электронного кошелька.</p> <p>Пример заполнения: 111222333444555_0987654321 где: «111222333444555» – номер эл.кошелька; «_» - разделитель (нижнее подчёркивание) 0987654321 – ИНН оператора электронного кошелька</p>	<pre>{ "wallet": "111222333444555_0987654321" }</pre>
extended.client_browser_info	Блок данных о браузере клиента используется в терминалах банка ПАО ВТБ	См. 3.3.3 (Применение Authorize3d) Инструкция по интеграции
qr_sbp	Поле "qrurl" генерируются если в поле "extended" был передан параметр qr_sbp:1	{"qr_sbp":1}
recipients	JSON массив, обязателен для заказов со сплитованием. Включает атрибуты:	{"recipients":[{"id":123, "amount":51.00}, {"id":345, "amount":49.00}]}
	id - идентификатор Поставщика услуг/merchant_id в системе банка amount - Сумма перечисления	
extended.layout	Формирование платежной формы с указанным шаблоном	{"layout" : "test_form"}

<p>messageExtension</p>	<p>JSON массив передается для карт МИР, если мсс=6050, 4814, 6538. Содержит информацию об идентификатор получателя денежных средств. Включает атрибуты: destWalletNum – Номер электронного кошелька получателя (мсс = 6050) destPhoneNum – Номер телефона для которого производится увеличение остатка средств клиента (мсс = 4814) destCardNum – Номер карты на которую будет совершаться перевод (мсс = 6538) destAccNum – БИК и номер банковского счета получателя перевода (мсс = 6538) Допускается заполнение только одного из атрибутов dest...</p>	<pre>"messageExtantion":{"destAccNum": "1111"}</pre>
<p>form_type</p>	<p>Параметр form_type используется для вывода определенного метода оплаты на платежной форме. Поддерживается массив из нескольких значений: card sbp payqr sberpay</p>	<pre>{"form_type": "sbp"} {"form_type": ["sbp", "card"]}</pre>