

PAYLER

ЗА ГРАНЬЮ СТАНДАРТОВ

Payler API

Описание интерфейса

Версия 1.1.9 от 27.09.2021

Оглавление

Оглавление.....	2
Изменения в документе.....	4
Термины и сокращения	5
Общие сведения	6
Введение	6
Типы операций.....	6
Процесс выполнения операций.....	7
Схема GATE	7
Схема MERCHANT.....	7
Способы оплаты	8
Статусы операций	9
Типы данных	9
Гарантированные уведомления	10
Фискальные электронные чеки.....	10
Управление учётными записями и картами пользователей	11
Сценарии сохранения карточных данных.....	11
Рекуррентные платежи	13
Сплитование платежей	16
3D-Secure	17
Платежи Apple Pay	20
Прием платежей в мобильных приложениях.....	20
Прием платежей на сайте мерчанта	20
Платежи GooglePay	23
Прием платежей в мобильных приложениях.....	23
Прием платежей на сайте мерчанта	25
Методы API: платежи.....	31
StartSession	31
PayGate	33
FindSession.....	34
PayMerchant	36
Block.....	41
3DSMethod	44
ThreeDsMethodComplete	45

ChallengeComplete	46
Send3DS	47
Charge	49
Retrieve	50
Refund	51
RepeatPay	51
GetTemplate	53
ActivateTemplate	54
GetStatus	55
GetAdvancedStatus	56
ApplePay	59
AppleValidateMerchant	60
GooglePay	61
Методы API: переводы	62
StartCreditSession	62
CreditGate	64
CreditMerchant	65
RepeatCredit	66
GetBalance	68
Методы API: фискальные чеки	69
Receipt	69
GetReceiptStatus	75
Методы API: сохранение данных карт	78
StartSaveCardSession	78
Save	79
PaySaveCard	80
GetStatusSaveCard	82
GetCardList	83
RemoveCard	84
SaveCard	85
CustomerRegister	86
CustomerUpdate	87
CustomerDelete	88
CustomerGetStatus	88
Ответы в случае неудачных запросов	89
Коды ошибок сервера	90

Изменения в документе

Версия	Дата	Изменения
1.1.0	19.02.2021	<p>Расширен список параметров и изменен Url запроса методов PayMerchant, Block и PaySaveCard</p> <p>Добавлены методы 3DSMethod, ThreeDsMethodComplete, ChallengeComplete</p> <p>Изменено описание процесса выполнения платежа для схемы MERCHANT</p> <p>Расширен и скорректирован раздел 3D-Secure</p>
1.1.1	12.03.2021	Увеличено максимально допустимое число символов для поля card_holder
1.1.2	26.03.2021	<p>Изменено описание параметров запроса PaySaveCard: исключен параметр card_id, скорректировано описание параметра customer_id, изменена обязательность параметров card_number, expired_year, expired_month</p> <p>Изменено описание RepeatPay: добавлена доступность по мапі, в параметры запроса добавлен параметр userdata</p>
1.1.3	05.04.2021	Изменены URL запросов для методов ThreeDsMethodComplete , ChallengeComplete , Send3DS , RepeatPay , PaySaveCard и SaveCard
1.1.4	27.04.2021	<p>Скорректировано описание recurrent_template_id в ряде методов.</p> <p>Изменена обязательность email в запросе StartCreditSession.</p> <p>Добавлена доступность GooglePay и ActivateTemplate для мапі.</p> <p>Изменены URL запросов SaveCard, GetBalance, RepeatPay, ApplePay, AppleValidateMerchant, GooglePay.</p> <p>Для поля phone в запросе Receipt установлено ограничение до 15 символов.</p> <p>В запрос StartSession и StartSaveCardSession добавлен параметр lifetime.</p> <p>Добавлена доступность FindSession для сгапі.</p> <p>Добавлена доступность ThreeDsMethodComplete для гапі.</p> <p>Метод отправки запроса Save изменен на GET</p>
1.1.5	09.06.2021	<p>Добавлен раздел Сплитование платежей</p> <p>В запросы StartSession, Block, PayMerchant, StartCreditSession и CreditMerchant добавлены опциональные параметры pay_page_param_addtoreport3 и pay_page_param_addtoreport4</p>
1.1.6	14.07.2021	<p>В описание кодов ошибок сервера добавлен код 1102.</p> <p>Из запросов исключен опциональный параметр airline_addendum.</p> <p>Скорректировано и дополнено описание 3DSMethod</p>
1.1.7	10.09.2021	В раздел Рекуррентные платежи добавлено примечание об ограничениях со стороны МПС
1.1.8	07.09.2021	Изменена обязательность параметра customer_id в методе GetCardList
1.1.9	27.09.2021	Изменена структура раздела Общие сведения

Термины и сокращения

Транзакция	— операция по переводу денег с одного счета на другой
Банк-эмитент	— банк, выдавший карту
Авторизация	— разрешение банка-эмитента на проведение платежа по карте
Карта, банковская карта	— платежная карта, привязанная к расчётному счету в банке
ККТ	— контрольно-кассовая техника, предназначенная для механизации кассовых операций, учета денежных поступлений, регистрации приобретения товара и печати кассового чека
Личный кабинет	— специальный раздел в авторизованной зоне сайта, содержащий данные, относящиеся к конкретному пользователю
МПС	— международная платежная система (VISA, MasterCard и т.п.)
Мерчант	— торгово-сервисное предприятие, подключенное к Payler
ОФД	— оператор фискальных данных, осуществляющее приём, обработку, хранение и передачу фискальных данных в ФНС России
Пользователь, покупатель	— лицо, осуществляющее оплату картой или получающее перевод на банковскую карту со счёта мерчанта
Фискальный чек, чек	— документ, формируемый зарегистрированным в налоговой службе расчётным кассовым аппаратом
Шлюз, Payler	— платёжный сервис Payler
PCI DSS	— единый международный стандарт безопасности данных индустрии платёжных карт

Общие сведения

Введение

Интерфейс Payler API позволяет выполнять стандартный набор операций электронной коммерции, а также перевод средств со счета мерчанта на карту пользователя по одной из выбранных схем взаимодействия:

1. С вводом данных на странице шлюза Payler (схема Gate).
2. С вводом данных банковских карт на странице мерчанта (схема Merchant).

В целях обеспечения безопасности платежей, МПС Visa и MasterCard требуют от стороны, оперирующей данными платежных карт, соответствовать требованиям стандарта PCI DSS. Для подтверждения соответствия стандарту мерчанту необходимо пройти верификацию, что может повлечь за собой дополнительные финансовые и организационные затраты.

Внимание

*Ответы API сервиса могут содержать незадокументированные поля.
При обработке ответов такие поля должны игнорироваться клиентским кодом.*

Типы операций

Платеж

Операция по переводу средств со счёта карты пользователя на счёт мерчанта.

- Одностадийный платёж — после авторизации сумма платежа блокируется на карте пользователя и автоматически списывается через сутки.
- Двухстадийный платёж — после авторизации сумма платежа блокируется на карте пользователя, но списание происходит только после подтверждения операции со стороны мерчанта.

Перевод

Операция по переводу средств со счёта мерчанта на счёт карты пользователя.

Процесс выполнения операций

Схема GATE

Платёж (gapi)

1. Мерчант вызывает метод [StartSession](#) для создания сессии платежа.
2. Мерчант отправляет пользователя на страницу оплаты ([PayGate](#)).
3. Пользователь вводит карточные данные, отправляет их в шлюз. По умолчанию пользователю доступно до 5 попыток оплаты. Количество попыток может быть индивидуально изменено по запросу мерчанта. При необходимости Payler перенаправляет пользователя на страницу банка-эмитента карты для ввода одноразового кода подтверждения ([3D-Secure](#)).
4. После завершения оплаты пользователь возвращается обратно на сайт мерчанта, на заранее заданный адрес.
5. При возврате пользователя мерчант вызывает метод [GetStatus](#) для проверки статуса заказа.

Перевод (cdapi)

1. Мерчант вызывает метод [StartCreditSession](#) для создания сессии перевода.
2. Мерчант отправляет пользователя на страницу перевода ([CreditGate](#)).
3. Пользователь вводит карточные данные.
4. После завершения перевода пользователя возвращают обратно на сайт мерчанта, на заранее заданный адрес.
5. При возврате пользователя мерчант вызывает метод [GetStatus](#) для проверки статуса перевода.

Схема MERCHANT

Платёж (mapi)

1. Пользователь вводит карточные данные на сайте мерчанта.
2. Мерчант, в зависимости от типа платежа, вызывает метод [PayMerchant](#) или [Block](#).
3. При необходимости прохождения дополнительной аутентификации с использованием 3D-Secure, в зависимости от используемого 3DS-протокола, мерчант выполняет дополнительные запросы [Send3DS](#), [ThreeDsMethodComplete](#), [ChallengeComplete](#).
4. После завершения оплаты мерчант вызывает метод [GetStatus](#) для проверки статуса заказа.

Перевод (старі)

1. Пользователь вводит карточные данные на сайте мерчанта.
2. Мерчант вызывает метод [CreditMerchant](#).
3. После завершения операции мерчант вызывает метод [GetStatus](#) для проверки статуса перевода.

В некоторых случаях пользователь может не вернуться на сайт мерчанта. Например, если он закрыл окно браузера сразу после оплаты или у него пропал доступ в Интернет. Чтобы всегда узнавать об изменении статуса заказа, мерчант может настроить получение гарантированных уведомлений ([callback](#)).

Способы оплаты

Способ	Описание
Card	Оплата банковской картой
Qiwi	Оплата через Qiwi-кошелёк
YandexMoney	Оплата через Яндекс-Деньги
WebMoney	Оплата через WebMoney
Beeline	Оплата с телефона (Билайн)
Megafon	Оплата с телефона (Мегафон)
Mts	Оплата с телефона (МТС)
Tele2	Оплата с телефона (Теле2)
AlfaClick	Оплата через интернет-банк (Альфа-банк)
SberOnline	Оплата через интернет-банк (Сбербанк)
PsbRetail	Оплата через интернет-банк (ПСБ)
Svyaznoy	Оплата в салоне Связного
Unistream	Оплата через Юнистрим
RussianPost	Оплата в отделениях Почты России
ApplePay	Оплата через ApplePay
GooglePay	Оплата через GooglePay

Статусы операций

Статус	Значение
Created	Платеж зарегистрирован в шлюзе, но его обработка не завершена
PreAuthorized3DS	Пользователь начал аутентификацию по протоколу 3D-Secure
Authorized	Средства заблокированы, но не списаны (двуухстадийный платеж)
Reversed	Средства на карте были заблокированы и полностью разблокированы
Charged	Денежные средства списаны с карты пользователя, платёж завершен успешно
Refunded	Успешно произведен полный возврат средств на карту пользователя
Rejected	Последняя операция по платежу отклонена
Pending	Операция находится в обработке. В этом случае желательно делать запросы статуса через временные промежутки в геометрической прогрессии, чтобы узнать окончательный статус платежа
Credited	Средства зачислены на карту пользователя, перевод завершен успешно

Типы данных

В описаниях методов API используются следующие обозначения типов данных:

Обозначение	Пояснение
A	Строка
A*	Строка с фиксированным количеством символов. Число указывает количество символов
A..*	Строка с ограничением количества символов. Число указывает максимальное допустимое количество символов
Ar	Массив
Ar..*	Массив с ограничением числа элементов. Число указывает максимальное допустимое количество элементов
B	Данные логического типа
D	Словарь
F	Дробное число
N	Целое число
N*	Целое число с фиксированным количеством символов. Число указывает количество символов

Гарантированные уведомления

Гарантированные уведомления (или — callback) служат для информирования мерчанта об успешных или неуспешных платежах и работают следующим образом: после изменения статуса заказа платежный шлюз посылает POST-запрос (Content-type: x-www-form-urlencoded) на заранее заданный URL.

В параметрах запроса передается *order_id*:

- если получен успешный код ответа (2xx), то обработка уведомления завершается;
- если получена ошибка, или сервер недоступен, то попытка повторяется через 10 секунд, 1 минуту, 15 минут, 1 час, 2 часа, 4 часа, 8 часов, 24 часа.

Для получения callback сервис мерчанта должен быть доступен по одному из следующих портов: 80, 433, 8080, 4443, 8443, 9443, 14405, 4405.

IP-адрес исходящего соединения: 178.20.235.180.

URL для отправки гарантированных уведомлений указывается мерчантом в Личном кабинете в разделе «Настройки».

Фискальные электронные чеки

Выдача чеков происходит следующим образом:

1. После совершения платежа мерчант вызывает метод [Receipt](#), в который передается вся информация, необходимая для выдачи чека. Чек поступает в обработку.
2. После завершения обработки чека, мерчант получает callback (POST-запрос), на URL, указанный в личном кабинете. В запросе передается параметр *order_id*.
3. После получения callback мерчант вызывает метод [GetReceiptStatus](#) и получает в ответ информацию по выданному чеку.

В личном кабинете сервиса АТОЛ ОНЛАЙН (а также — в личном кабинете ОФД) доступна статистика и отчетность по выданным чекам. После регистрации чека в ККТ, ОФД самостоятельно проводит рассылку чеков плательщикам по указанным контактным данным (e-mail, sms).

Управление учётными записями и картами пользователей

Для удобства постоянных пользователей мерчант может сохранить их карточные данные для последующих платежей. Данные платежной карты пользователя могут вводиться как на стороне мерчанта, так и на веб-странице защищенного шлюза Payler. В первом случае мерчант должен соответствовать требованиям стандарта безопасности PCI DSS.

Описание методов, обеспечивающих управление учетными записями и платёжными картами пользователей, см. в разделе [Методы API: сохранение данных карт](#).

Сценарии сохранения карточных данных

Привязка карт выглядит следующим образом:

1. Мерчант создает пользователя в системе Payler метод [CustomerRegister](#).
2. Мерчант привязывает карты к созданному пользователю одним из способов:
 - Во время проведения платежа; метод [StartSession](#).
 - С проверкой суммы и автоматическим возвратом (разблокировкой) средств; метод [StartSaveCardSession](#) (схема Gate) или [PaySaveCard](#) (схема Merchant).
3. При последующих оплатах мерчант указывает идентификатор пользователя (*customer_id*), и на странице оплаты пользователь сможет выбрать одну из своих ранее сохраненных карт или указать новую карту.
4. Удаление карты ([RemoveCard](#)) и удаление пользователя ([CustomerDelete](#)) необратимы. При удалении пользователя все привязанные к нему карты будут также удалены.

Для последующих операций Payment (платеж) требуется хранить и использовать **reccurent_template_id**. Для последующих операций Credit (перевод) требуется хранить и использовать **card_id**

Сохранение карты при оплате

Сценарий выполнения:

1. Мерчант вызывает метод [StartSession](#) с ключевыми параметрами `reccurrent=true` и `customer_id`.
2. Мерчант отправляет пользователя на страницу оплаты (метод [PayGate](#)).
3. Пользователь вводит и отправляет карточные данные.
4. После завершения оплаты пользователь возвращается на сайт мерчанта, на заранее заданный адрес. Происходит отправка [callback](#) (опционально).
5. При возврате пользователя или получении callback мерчант может вызвать метод [GetAdvancedStatus](#) для проверки статуса заказа и получения `card_id/reccurent_template_id`

Если по платежу отсутствует 3DS (карта не вовлечена в 3D-Secure), карта не сохраняется; Если платеж отклоняется по балансу или проходит успешно, то для последующего списания (метод [RepeatPay](#)) создается рекуррентный шаблон, и карта сохраняется для операций

выдачи средств (метод [RepeatCredit](#)). Данные шаблона отправляется исключительно в callback.

Сохранение карты с проверкой суммы и автоматическим возвратом (разблокировкой)

В данном сценарии проводится блокировка или списание денежных средств в размере 1 руб. с последующей автоматической отменой операции. Тип выполняемой операции (блокировка/списание) зависит от настроек процессинга.

Сценарий выполнения для схемы Gate:

1. Мерчант вызывает метод [StartSaveCardSession](#) для создания сессии сохранения карты.
2. Мерчант отправляет пользователя на страницу сохранения карты (метод [Save](#)).
3. Пользователь вводит и отправляет карточные данные.
4. После завершения оплаты пользователь возвращается на сайт мерчанта, на заранее заданный адрес. Происходит отправка callback (опционально).
5. При успешном проведении проверочного списания производится автоматический возврат средств.
6. При возврате пользователя или получении callback мерчант может вызвать метод [GetAdvancedStatus](#) для проверки статуса заказа и получения card_id/reccurent_template_id.

Сценарий выполнения для схемы Merchant:

1. Мерчант вызывает метод [PaySaveCard](#).
2. Пользователь вводит и отправляет карточные данные.
3. При успешном проведении проверочного списания производится автоматический возврат или отмена блокировки средств.
4. Происходит отправка callback (опционально).
5. При получении callback мерчант может вызвать метод [GetAdvancedStatus](#) для проверки статуса заказа и получения card_id/reccurent_template_id.

Особенности:

- Если по платежу отсутствует 3DS (карта не вовлечена в 3D-Secure), карта не сохраняется;
- Если платеж прошел успешно или отклонился по балансу, то карта сохраняется и пользователю демонстрируется форма с информацией о том, что карта сохранена успешно. Шаблон формы может быть кастомизирован.
- Если операция завершилась успешно, то автоматически происходит возврат (разблокировка) средств.
- Если платеж отклоняется по балансу или проходит успешно, то для последующих платежей создается рекуррентный шаблон и карта сохраняется для операций выдачи средств (Credit). Данные шаблона (_template_id) отправляются исключительно в callback.

Примечание

Для сохранения карточных данных с последующей возможностью списания или выдачи средств рекомендуется использовать сценарий с вызовом метода [StartSaveCardSession](#).

В этом случае, и при успешном совершении тестовой оплаты (блокировки), и при отклонении по балансу карточные данные будут сохранены и для пользователя отобразится страница с информацией об успешном сохранении карты.

Рекуррентные платежи

Рекуррентные (регулярные, повторяющиеся) платежи — это платежи, которые не требуют повторного ввода реквизитов карты. Пользователь единожды производит оплату и соглашается с условиями регулярного списания; последующее списание происходит без его участия. При осуществлении первого платежа в серии, в базе данных регистрируется шаблон рекуррентных платежей. Шаблону присваивается уникальный идентификатор, который сообщается мерчанту. При осуществлении повторного платежа мерчант выполняет запрос с указанием полученного идентификатора шаблона.

Первый платеж выполняется с вводом всех реквизитов карты, включая код подлинности карты (CVV2/CVC2) и прохождение авторизации по протоколу 3D-Secure. Последующие платежи выполняются без ввода реквизитов карты и без участия владельца карты.

Принципы работы

При выполнении платежа мерчант может передать специальный параметр, который будет означать, что данный платеж — это первый платеж в очереди повторяющихся платежей. Если оплата произведена успешно, мерчант получит идентификатор созданного шаблона. Повторные платежи будут выполняться с использованием этого идентификатора. Срок действия шаблона задается равным сроку действия банковской карты, к которой он привязан. Информацию о зарегистрированных шаблонах рекуррентных платежей можно получить в результате выполнения запроса [GetTemplate](#); при этом требуется указать идентификатор шаблона.

Активация или деактивация шаблона выполняется в результате запроса [ActivateTemplate](#); при этом требуется указать идентификатор шаблона.

Автоматическое удаление шаблонов через Payler API недоступно. Удалить шаблон можно по запросу к технической службе Payler.

Суммы повторных платежей могут быть как больше, так и меньше суммы первоначального платежа. Первоначальный платеж можно отменить (вернуть деньги пользователю), но возможность выполнять повторные платежи останется.

Первоначальный платеж в серии может быть как одностадийным, так и двухстадийным. В случае двухстадийного платежа шаблон создается при блокировании средств, но становится активным только после того, как успешно выполнено списание заблокированных средств. Повторные платежи всегда — одностадийные.

Сценарии проведения платежа

После успешного проведения платежа мерчант получает идентификатор созданного шаблона рекуррентных платежей (параметр *recurrent_template_id*) в callback.

Обязательным условием является использование протокола HTTPS для [callback](#).

Предварительно мерчанту необходимо выполнить настройку callback для триггера charge или block. Настроить callback можно самостоятельно в личном кабинете мерчанта или обратившись в службу поддержки Payler.

Этапы сценария

1. Выполнение платежа

Создать платежную сессию (запрос [StartSession](#)) с указанием значения параметра *recurrent = true*. Перенаправить пользователя на запрос [PayGate](#). Или (только для схемы Merchant) — выполнить запрос [PayMerchant](#) или [Block](#) с указанием параметра *recurrent = true*.

2. Сохранение шаблона рекуррентных платежей

После проведения успешного платежа Payler отправляет мерчанту идентификатор созданного шаблона в параметре *recurrent_template_id* с помощью callback.

После получения callback требуется запросить статус платежа (запрос [GetStatus](#)).

Это необходимо, чтобы убедиться, что платеж прошел успешно.

В случае успешного списания мерчант сохраняет полученный *recurrent_template_id*.

Если статус платежа отличен от charged/authorized, рекуррентный шаблон надо считать недействительным. Требуется обратиться в службу поддержки.

3. Выполнение повторного платежа

Для проведения повторного платежа необходимо выполнить запрос [RepeatPay](#), с указанием полученного параметра *recurrent_template_id*.

Внимание

Правила МПС ограничивают число попыток рекуррентного списания средств.

- *При получении отказа по рекуррентному платежу по карте VISA повторный запрос на авторизацию можно направлять не более 4 раз в течение 16 календарных дней;*
- *При получении отказа по рекуррентному платежу по карте MasterCard повторный запрос на авторизацию можно направлять не чаще 1 раза в день в течение 31 дня.*

Нарушение мерчантом правил попыток рекуррентного списания средств может понести за собой оспаривание платежей (chargeback) в пользу плательщика.

4. Запрос статуса повторного платежа

Необходимо получить статус повторного платежа, используя запрос [GetStatus](#), с указанием параметров *key*, *order_id*.

В процессе интеграции мерчанту требуется сообщить службе технической поддержки Payler о необходимости подключения и настройки рекуррентных платежей на его аккаунте.

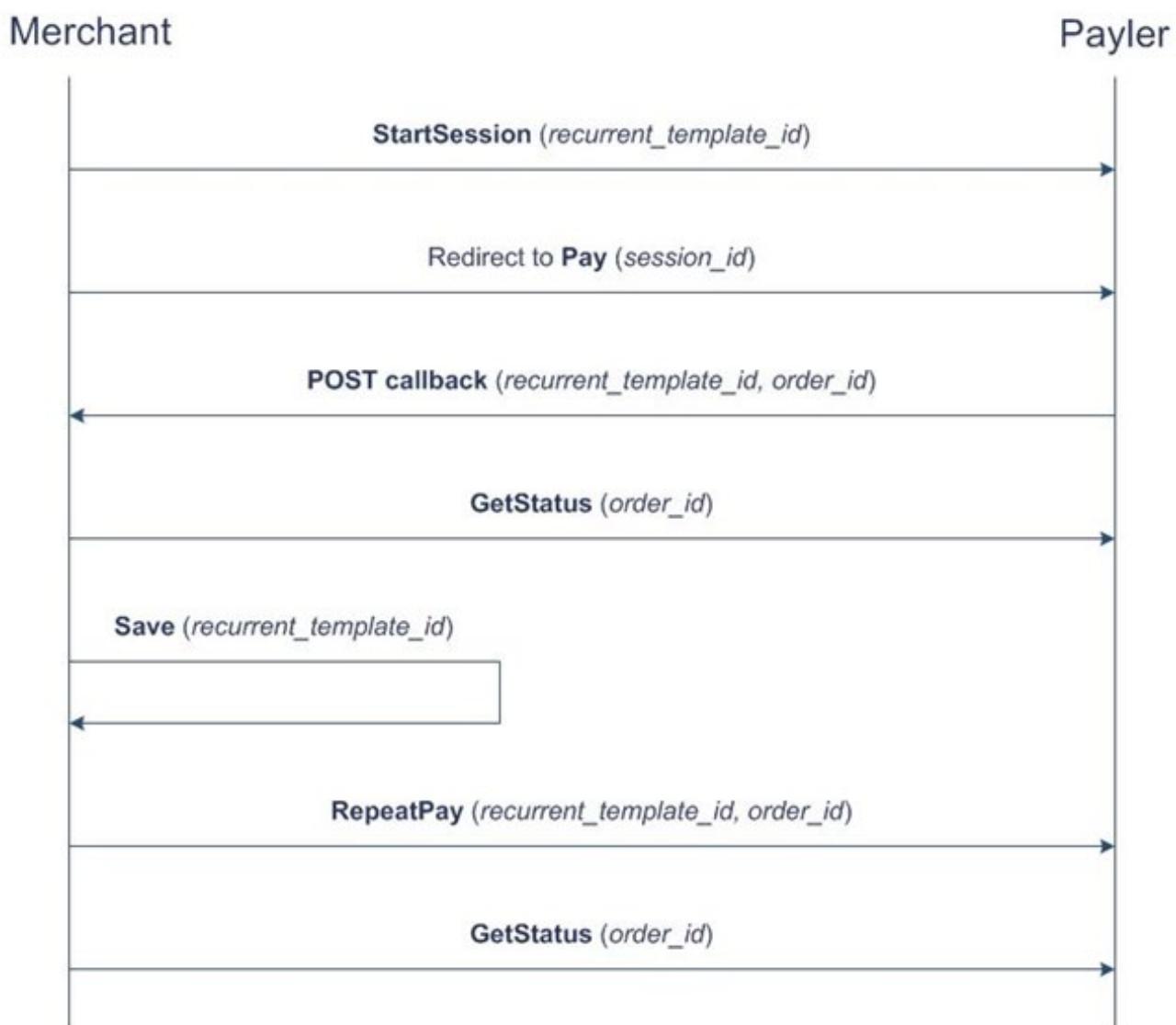


Схема проведения платежа с созданием шаблона рекуррентных платежей

Сплитование платежей

Если сумма платежа или перевода должна быть разделена между различными партнерами / клиентами ТСП, мерчант может передавать в запросе дополнительные параметры, содержащие данные о суммах перечислений и идентификаторах всех получателей. Это позволяет проводить транзакции в адрес нескольких получателей в рамках одной операции.

Доступность сплитования для мерчанта зависит от конкретного банка-эквайера.

Для уточнения возможности использования данного функционала и регистрации реквизитов получателей для сплитования необходимо связаться со службой технической поддержки Payler.

Принцип работы

1. Мерчант присыпает реквизиты партнеров, на которые будет проводиться сплитование дебетовых операций или дополнительная выплата кредитных операций.
2. После регистрации реквизитов мерчант получает ID получателей, которые закрепляются за этими реквизитами. Необходимые реквизиты: наименование организации, ИНН, БИК, Р/С.
3. Для запросов [StartSession](#) (при использовании [gapi](#)) или [PayMerchant](#), [Block](#) (при использовании [mapi](#)) необходимо передать дополнительные параметры следующим образом:
 - pay_page_param_addtoreport3 — суммы зачисления частей платежа для каждого получателя через запятую. Например: «30000,50000».
 - pay_page_param_addtoreport4 — ID получателей через запятую.
Например: «322_1,322_2».
4. Для запросов [StartCreditSession](#) (при использовании [cqapi](#)) или [CreditMerchant](#) (при использовании [cmapi](#)) необходимо передать дополнительные параметры следующим образом:
 - pay_page_param_addtoreport3 — суммы переводов с р/с мерчанта для каждого получателя через запятую. Например: «3000000,7800».
 - pay_page_param_addtoreport4 — ID получателей через запятую.
Например: «322_3, 322_4».

3D-Secure

3D-Secure является XML-протоколом, который используется для двухфакторной аутентификации пользователя в качестве дополнительного уровня безопасности для онлайн-кредитных и дебетовых карт. Название 3DS происходит от «3 Domains» (три домена), так как в проверке платежа по данному протоколу участвуют организации на трех доменах:

1. Домен эмитента (пользователь и банк-эмитент),
2. Домен эквайера (банк-эквайер и ТСП),
3. Домен взаимодействия (МПС).

Существует две версии протокола 3D-Secure — 3DS 1.0 и более современный и внедряемый сейчас банками-эмитентами 3DS 2.0. Переход на использование версии 2.0 осуществляется постепенно, так как часть эмитентов еще не поддерживает 3DS 2.0. При невозможности проведения аутентификации по новому протоколу аутентификация выполняется по 3DS 1.0.

Принципиальное отличие новой версии протокола состоит:

- в возможности осуществить аутентификацию без непосредственного участия пользователя (frictionless flow),
- в использовании 3DS Method для проведения аутентификации с проверкой держателя карты. В этом случае эмитент получает доступ к скрытому iframe в браузере пользователя и самостоятельно производит сбор данных о браузере, требуемых для аутентификации.

При поступлении запроса на проведение платежа или сохранение карты, для дальнейшего выполнения операции Payler на своей стороне выполняет проверку вовлеченности карты в 3D-Secure и определение поддерживаемой версии протокола.

3DS 1.0

Краткий сценарий аутентификации по версии 3DS 1.0 (3DS 1.0 Flow):

Держатель карты совершает оплату на сайте ТСП, вводит данные карты и направляется на страницу банка-эмитента, для ввода уникального кода.

Код отправляется пользователю банком-эмитентом по другому каналу (например, в sms). Если код введен правильно, банк-эмитент сообщает об успешном завершении проверки и операция успешно завершается.

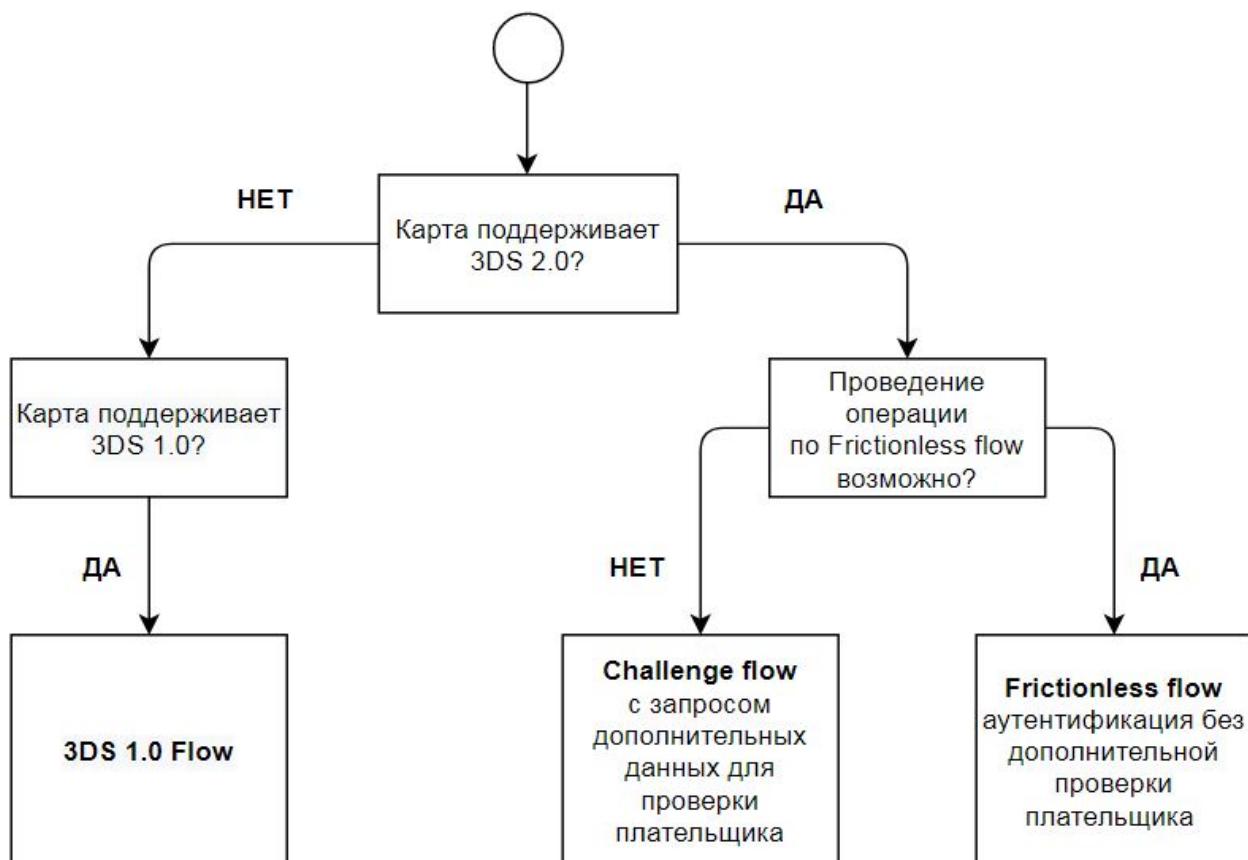
3DS 2.0

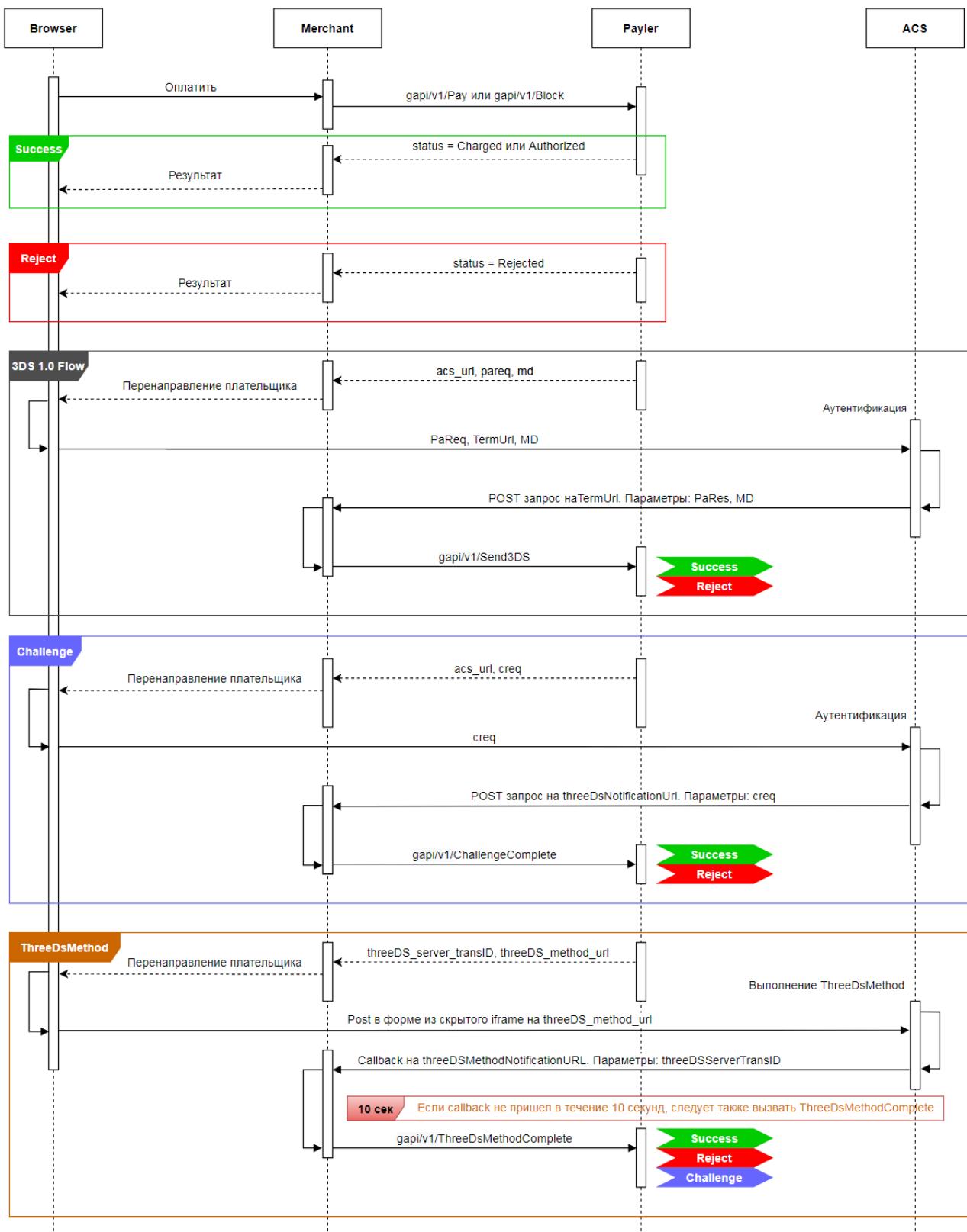
3DS 2.0 позволяет эмитенту использовать несколько режимов проверки подлинности клиента, в том числе с вводом одноразового кода, отправляемого эмитентом в sms. Но в отличие от версии 3DS 1.0 ввод пароля не является обязательным. Решение о подтверждении операции принимается на основании данных об устройстве, с которого совершается платеж, настроек браузера, IP-адреса, e-mail и др. Эмитент карты использует

этую информацию для оценки уровня риска и необходимости выполнения дополнительной проверки пользователя. В зависимости от результатов анализа степени риска эмитент принимает решение о дальнейшем проведении аутентификации:

- **Frictionless flow** — если риск мошенничества ниже заданного порогового значения, то эмитент не запрашивает дополнительную проверку плательщика и считает, что проверка подлинности владельца карты пройдена. В этом случае этап «ручной проверки» с вводом плательщиком дополнительного кода исключается.
- **Challenge flow** — если риск мошенничества выше заданного порогового значения, выполняется дополнительная проверка держателя карты аналогично 3DS 1.0.

Сценарии проведения операций с 3DS





Возможные сценарии 3D-Secure аутентификации

Платежи Apple Pay

Описание ниже относится к предоставлению услуги оплаты с помощью Apple Pay в мобильном приложении или на веб-сайте мерчанта. Для проведения платежа с использованием Apple Pay через платежную форму Payler никаких дополнительных действий по интеграции не требуется; для включения этого способа оплаты достаточно связаться со службой технической поддержки Payler.

Прием платежей в мобильных приложениях

Регистрация мерчанта

Для использования технологии Apple Pay мерчанту необходимо зарегистрировать Merchant ID и сформировать платежный сертификат:

1. Payler создает CSR и отправляет его мерчанту.
2. Мерчант загружает сертификат в свой аккаунт разработчика Apple, как описано в [документации Apple](#).
3. Мерчант передает в Payler полученный сертификат и Merchant ID.

Проведение оплаты

Схема оплаты включает 3 этапа:

1. Проверку совместимости устройства — если устройство поддерживается, то нужно показать покупателю кнопку Apple Pay.
2. Подтверждение платежа — подтверждение покупателем оплаты отпечатком пальца.
3. Обработку платежа — после подтверждения Apple формирует зашифрованный токен, который необходимо передать в API Payler.

Для этого требуется:

- вызвать метод [StartSession](#) для инициализации сессии платежа.
- после подтверждения платежа пользователем параметр [paymentData](#) из объекта [PKPaymentToken](#) передать в Payler с помощью метода [ApplePay](#); в ответ придет информация о статусе платежа.
- вызвать в приложении *completion*, передав в него статус платежа.

Интеграция ApplePay в мобильных приложениях описана в [документации Apple](#).

Прием платежей на сайте мерчанта

Страница с кнопкой Apple Pay должна отдаваться с через HTTPS с использованием валидного SSL-сертификата и протокола TLS 1.2.

Необходимо:

1. Получить от Payler файл apple-developer-merchantid-domain-association,

2. Разместить полученный файл на сервере мерчанта по адресу: <https://адрес.сайта/.well-known/apple-developer-merchantid-domain-association>.

Проведение оплаты

Схема оплаты включает 4 этапа:

1. Проверку совместимости устройства — если устройство поддерживается, нужно показать покупателю кнопку Apple Pay.
2. Валидацию мерчанта — необходимо выполнить запрос [AppleValidateMerchant](#) для валидации мерчанта в Apple.
3. Подтверждение платежа — подтверждение покупателем оплаты отпечатком пальца.
4. Обработку платежа — после подтверждения Apple формирует зашифрованный токен, который необходимо передать в API Payler с использованием метода [ApplePay](#).

Отображение кнопки ApplePay

Пример CSS и HTML для добавления кнопки Apple Pay на странице приведен в [документации Apple](#). Изначально кнопка Apple Pay должна быть невидима (display: none), а отображаться, только если есть возможность оплатить с помощью Apple Pay с данного устройства.

Обработка оплаты в JavaScript

Сначала требуется проверить возможность оплаты через Apple Pay на данном устройстве:

```
var promise = ApplePaySession.canMakePaymentsWithActiveCard(merchantIdentifier);
promise.then(function (canMakePayments) {
  if (canMakePayments) {
    // если возможно — отобразить кнопку
    document.getElementById("apple-pay-button").style.display = "block";
  }
});
```

При нажатии на кнопку создать сессию на оплату:

```
document.getElementById("apple-pay-button").onclick = function(evt) {
  var paymentRequest = {
    currencyCode: 'RUB',
    countryCode: 'RU',
    total: {
      label: 'Описание продукта',
      amount: 100.00 // цена в рублях
    },
    supportedNetworks: ['masterCard', 'visa'],
    merchantCapabilities: [ 'supports3DS', 'supportsCredit', 'supportsDebit' ]
  };
  var session = new ApplePaySession(2, paymentRequest);
```

Далее необходимо добавить обработчики событий в объект `session`:

Валидация мерчанта:

```
session.onvalidatemerchant = function (event) {
    // Нужно отправить event.validationURL на свой сервер, откуда вызвать
    // метод Payler AppleValidateMerchant, а полученный ответ передать в
    // метод session.completeMerchantValidation

    session.completeMerchantValidation(JSON.parse(response.validation_object));
}
```

Обработка платежа:

```
session.onpaymentauthorized = function (event) {
    // необходимо отправить event.payment.token.paymentData на свой сервер, откуда
    // вызвать метод Payler StartSession, а затем ApplePay.
    // Результат проведения платежа нужно вернуть обратно на клиент.

    // Если платёж успешен, вызываем:
    session.completePayment(ApplePaySession.STATUS_SUCCESS);

    // Если платёж неуспешен, вызываем:
    session.completePayment(ApplePaySession.STATUS_FAILURE);
}
```

Отображение формы Apple Pay

```
session.begin();
```

Более подробно методы и события ApplePaySession описаны в [документации Apple](#).

Платежи GooglePay

Описание ниже относится к предоставлению услуги оплаты с помощью Google Pay в мобильном приложении или на веб-сайте мерчанта. Для проведения платежа с использованием Google Pay через платежную форму Payler никаких дополнительных действий по интеграции не требуется; для включения этого способа оплаты достаточно связаться со службой технической поддержки Payler.

В случае использования Google Pay мерчант обязан соблюдать положения следующих правил: <https://payments.developers.google.com/terms/sellertos>

Документация к API: <https://developers.google.com/pay/api/web>

Правила использования бренда: <https://developers.google.com/pay/api/web/guides/brand-guidelines>

Контрольный список интеграции: <https://developers.google.com/pay/api/web/guides/test-and-deploy/integration-checklist>

Мерчант может самостоятельно выбрать типы карт (методов аутентификации), по которым возможна оплата в Google Pay.

Доступны два метода аутентификации:

1. **CRYPTOGRAM 3DS** — для токенизованных карт, данные которых хранятся в виде токенов на устройстве клиента и доступных только на том устройстве, на котором карта была добавлена в приложение Google Pay.
2. **PAN ONLY** — для нетокенизованных карт, данные которых хранятся в аккаунте Google и доступных на любом устройстве клиента.

Тип аутентификации указывается в настройках при интеграции с Google Pay в свойстве allowedAuthMethods.

Внимание

Метод PAN ONLY доступен не для всех мерчантов.

Возможность подключения метода необходимо уточнять индивидуально.

Прием платежей в мобильных приложениях

Настройка Google Pay API

Заполнение файла [Constants.java](#):

```
1. package com.google.android.gms.samples.wallet;
2.
3. import com.google.android.gms.wallet.WalletConstants;
4. import java.util.Arrays;
```

```

5. import java.util.HashMap;
6. import java.util.List;
7.
8. public class Constants {
9.
10.    public static final int PAYMENTS_ENVIRONMENT =
11.        WalletConstants.ENVIRONMENT_TEST; // WalletConstants.ENVIRONMENT_PRODUCTION для
12.        production среды.
13.    public static final List<String> SUPPORTED_NETWORKS = Arrays.asList(
14.        "MASTERCARD",
15.        "VISA");
16.    public static final List<String> SUPPORTED_METHODS =
17.        Arrays.asList(
18.            "PAN_ONLY",           //Оплата по карточным данным из аккаунта Google.
19.            "CRYPTOGRAM_3DS"); //Оплата только при наличии на устройстве Google Pay.
20.        Является более безопасным способом. Можно использовать как оба способа, так и любой один
21.        из них.
22.    public static final String CURRENCY_CODE = "RUB"; // Или USD, EUR. Должна
23.        совпадать с валютой в StartSession.
24.    public static final String PAYMENT_GATEWAY_TOKENIZATION_NAME = "payler";
25.    public static final HashMap<String, String>
26.        PAYMENT_GATEWAY_TOKENIZATION_PARAMETERS =
27.        new HashMap<String, String>() {
28.            put("gateway", PAYMENT_GATEWAY_TOKENIZATION_NAME);
29.            put("gatewayMerchantId", "Ваш идентификатор в системе Payler");
30.        };
31.    private Constants() {}
32.}
```

В файле [PaymentsUtil.java](#) должен быть установлен тип токенизации “PAYMENT_GATEWAY”.

Проведение платежа

1. Вызвать метод [StartSession](#) для инициализации сессии платежа.
2. В обработчике нажатия кнопки Google Pay в случае успеха вызывать метод [GooglePay](#), передав:
 - session_id из ответа [StartSession](#),
 - строку "tokenizationData.token" из полученного от Google объекта PaymentData (google_pay_token),
 - почтовый адрес покупателя (email).

Пример токена:

```
{
  "signature": "MEYCIQCKSQPbDN51Kh1Y+16W3A5fdW3Patf0hoBrXmTudQZSSQIhAMX8G54mswpbw07t1adHhNfZAkWUvQ6YyQP8
  Zzhc68pR",
  "intermediateSigningKey": {
    "signedKey": "
{\\"keyValue\\":\\"MFkwEwYHKoZIZj0CAQYIKoZIZj0DAQcDQgAEzHwfz1Vq1ZIf9fiZVyBI33CnXRwBP2hJ1CYv
"
  }
}
```

```

zx5x6adfBXfYv0krbQU1bEv9DNKwStI1RKhzQhIdεJZBGF196w\\u003d\\u003d\", \"keyExpiration\": \"15
51450347229\" }",
    "signatures": [
        "MEYCIQC6Cws1WBByrGnoijw0MWcjvq16aYCL+JH/+5Y3DAMbJiwIhAL6+RWUJo+j9cqaP71UsZs18GPVU0kq/CLOB
0Qb8rBgr"
    ]
},
"protocolVersion": "ECv2",
"signedMessage":
" \\"encryptedMessage\\": \"WBGMAnIuhwyGROX6c4q/Ux6Gupm/US2FVnq7wKZEp7C6E50kiHqv/2vd+PW4/XVY
1V1Khh+PoT6RY66Q3OX/Q2apalv+4OCxWJhlsP4sZ0NZCCiJEfWY5P9ayjNbzJJnoJerfwykn2cDITHHjerHtNOBM
UkqCFkW7Yj0X5uZTfx1Alr2k4knLWtMrlfHuh9RHiaEi/pcppWerYmNPrlzU615ypqinwtifq+iVldnz2oYJpDzQy
ZukFU51Xu5BQhzHUtqly9Y1Lt1hCfCeLY4fqrz/V10oiXmgeY9gUrJOy6zzSdAq5VsdiZlUCbsP21ThXG9Hg/ZP
Q7o0fxdSnLI3BvkjFLvFtLgDtnk9o8Di9kx2intm/5YJS0RB1wOfjicn5JzwuDiur/RXYMPjhAh0L+ZUts3knWMN1
giVXAuEPwRm5oU/2RiThV3du3BNJSQbybfIvSIhH12Pwua1tFLRKzESBnhfrP/OFN52PaazQQ4Jv7PG+tpVuvWEer
difoGuVXQw89vloCYKYBBEd1NG85DSkMYq/hAfmsgFrDrs7spA1Sb26s5oJR0mL76x90X8\\u003d\", \"ephemer
alPublicKey\": \"BLVkBHOrk1we3aAqbHeMfnL3dfGUmM9MQJ8CJtLwQn6KUnpySSSY0os+PHDbqfIHVVa8jqzgx
1gynofubTsqqTA\\u003d\", \"tag\": \"51c6cXbwEd9obykJ+qbAj+O6fHFwP85LGnihp4//zDg\\u003d\" } "
}

```

Прием платежей на сайте мерчанта

Сайт мерчанта должен работать по HTTPS и поддерживать протокол TLS версии 1.2. Домен сайта необходимо предварительно зарегистрировать и подтвердить в Google, заполнив [форму регистрации](#). После отправки формы представитель Google связывается с мерчантом для консультации по дальнейшим действиям.

Интеграция включает использование серверной и клиентской (javascript) части. На клиенте выполняется проверка совместимости устройства и получение платежных данных, на сервере отправляется запрос на проведение платежа.

Настройки

Пример js кода:

- Параметр gateway в скрипте имеет постоянное значение "payler";
- Значение параметра gatewayMerchantId мерчант получает от Payler;
- Значение параметра merchantId мерчант получает от Google,
- Значение параметра merchantName соответствует названию интернет-магазина, указанному при регистрации в Google.

```

/**
 * Define the version of the Google Pay API referenced when creating your
 * configuration
 *
 * @see {@link
https://developers.google.com/pay/api/web/reference/object#PaymentDataRequest|apiVersion
in PaymentDataRequest}
 */

```

```

var baseRequest = {
  apiVersion: 2,
  apiVersionMinor: 0,
};

var allowedCardNetworks = ['MASTERCARD', 'VISA'];
// CRYPTOGRAM_3DS - Оплата только при наличии на устройстве Google Pay. Является более
// безопасным способом. На текущий момент работает только в браузере Chrome на Android.
// PAN_ONLY - Оплата по карточным данным из аккаунта Google. Доступен в большинстве
// современных браузеров.
// Можно использовать как оба способа, так и любой один из них.
var allowedCardAuthMethods = ["PAN_ONLY", "CRYPTOGRAM_3DS"];
var tokenizationSpecification = {
  type: 'PAYMENT_GATEWAY',
  parameters: {
    gateway: 'payler',
    gatewayMerchantId: ' ваш Id в системе Payler',
  },
};

/**
 * Describe your site's support for the CARD payment method and its required
 * fields
 *
 * @see {@link
https://developers.google.com/pay/api/web/reference/object#CardParameters|CardParameters
 *}
var baseCardPaymentMethod = {
  type: 'CARD',
  parameters: {
    allowedAuthMethods: allowedCardAuthMethods,
    allowedCardNetworks: allowedCardNetworks,
  },
};

/**
 * Describe your site's support for the CARD payment method including optional
 * fields
 *
 * @see {@link
https://developers.google.com/pay/api/web/reference/object#CardParameters|CardParameters
 *}
var cardPaymentMethod = Object.assign({}, baseCardPaymentMethod, {
  tokenizationSpecification: tokenizationSpecification,
});

var paymentsClient = null;

/**
 * Configure your site's support for payment methods supported by the Google Pay
 * API.
 *
 * Each member of allowedPaymentMethods should contain only the required fields,
 * allowing reuse of this base request when determining a viewer's ability
 * to pay and later requesting a supported payment method
 *
 * @returns {object} Google Pay API version, payment methods supported by the site
 */
function getGoogleIsReadyToPayRequest() {
  return Object.assign({}, baseRequest, {
    allowedPaymentMethods: [baseCardPaymentMethod],
  });
}

```

```

}

/**
 * Configure support for the Google Pay API
 *
 * @see {@link
https://developers.google.com/pay/api/web/reference/object#PaymentDataRequest|PaymentData
Request}
 * @returns {object} PaymentDataRequest fields
 */
function getGooglePaymentDataRequest() {
    var paymentDataRequest = Object.assign({}, baseRequest);
    paymentDataRequest.allowedPaymentMethods = [cardPaymentMethod];
    paymentDataRequest.transactionInfo = getGoogleTransactionInfo();
    paymentDataRequest.merchantInfo = {
        // @todo a merchant ID is available for a production environment after approval by
        Google
        // See {@link https://developers.google.com/pay/api/web/guides/test-and-
        deploy/integration-checklist|Integration checklist}
        merchantId: 'Ваш Id GooglePay',
        merchantName: 'Ваше наименование',
    };
    return paymentDataRequest;
}

/**
 * Return an active PaymentsClient or initialize
 *
 * @see {@link
https://developers.google.com/pay/api/web/reference/client#PaymentsClient|PaymentsClient
constructor}
 * @returns {google.payments.api.PaymentsClient} Google Pay API client
 */
function getGooglePaymentsClient() {
    if (paymentsClient === null) {
        paymentsClient = new google.payments.api.PaymentsClient({
            environment: 'TEST', // Для продуктовой среды д.б. PRODUCTION
        });
    }
    return paymentsClient;
}

/**
 * Initialize Google PaymentsClient after Google-hosted JavaScript has loaded
 *
 * Display a Google Pay payment button after confirmation of the viewer's
 * ability to pay.
 */
function onGooglePayLoaded() {
    {
        var paymentsClient = getGooglePaymentsClient();

        paymentsClient
            .isReadyToPay(getGoogleIsReadyToPayRequest())
            .then(function(response) {
                if (response.result) {
                    addGooglePayButton();
                    // @todo prefetch payment data to improve performance after confirming site
                    functionality
                    // prefetchGooglePaymentData();
                }
            })
    }
}

```

```

        .catch(function(err) {
            // show error in developer console for debugging
            console.error(err);
        });
    }

/**
 * Add a Google Pay purchase button alongside an existing checkout button
 *
 * @see {@link
https://developers.google.com/pay/api/web/reference/object#ButtonOptions|Button options}
 * @see {@link https://developers.google.com/pay/api/web/guides/brand-guidelines|Google
Pay brand guidelines}
 */
function addGooglePayButton() {
    var paymentsClient = getGooglePaymentsClient();
    var button = paymentsClient.createButton({onClick: onGooglePaymentButtonClicked});
    var changeBlock = document.querySelector('.change-js');
    document.getElementById('container').appendChild(button);
    changeBlock.style.display = 'block';
}

/**
 * Provide Google Pay API with a payment amount, currency, and amount status
 *
 * @see {@link
https://developers.google.com/pay/api/web/reference/object#TransactionInfo|TransactionInfo}
 *
 * @returns {object} transaction info, suitable for use as transactionInfo property of
PaymentDataRequest
 */
function getGoogleTransactionInfo() {
    return {
        currencyCode: 'RUB', //Валюта платежа
        totalPriceStatus: 'FINAL',
        // set to cart total
        totalPrice: '11.20', //Сумма платежа
    };
}

/**
 * Prefetch payment data to improve performance
 *
 * @see {@link
https://developers.google.com/pay/api/web/reference/client#prefetchPaymentData|prefetchPa
ymentData()}
 */
function prefetchGooglePaymentData() {
    var paymentDataRequest = getGooglePaymentDataRequest();
    // transactionInfo must be set but does not affect cache
    paymentDataRequest.transactionInfo = {
        totalPriceStatus: 'NOT_CURRENTLY_KNOWN',
        currencyCode: 'RUB',
    };
    var paymentsClient = getGooglePaymentsClient();
    paymentsClient.prefetchPaymentData(paymentDataRequest);
}

/**
 * Show Google Pay payment sheet when Google Pay payment button is clicked
 */

```

```

function onGooglePaymentButtonClicked() {
    var paymentDataRequest = getGooglePaymentDataRequest();
    paymentDataRequest.transactionInfo = getGoogleTransactionInfo();

    var paymentsClient = getGooglePaymentsClient();
    paymentsClient
        .loadPaymentData(paymentDataRequest)
        .then(function(paymentData) {
            // handle the response
            processPayment(paymentData);
        })
        .catch(function(err) {
            // show error in developer console for debugging
            console.error(err);
        });
}

function sendPaymentData(paymentToken) {
    return new Promise(function(resolve, reject) {
        var xhr = new XMLHttpRequest();
        xhr.onload = function() {
            var data = JSON.parse(this.responseText);
            resolve(data);
        };
        xhr.onerror = reject;
        xhr.open(
            'POST',
            'https://secure.payler.com/gapi/GooglePay?session_id=' + // https://secure.payler.com
для тестовой среды
            encodeURIComponent(window.params_payler.session_id) +
            '&google_pay_token=' +
            encodeURIComponent(paymentToken)
        );
        xhr.send();
    });
}

/**
 * Process payment data returned by the Google Pay API
 *
 * @param {object} paymentData response from Google Pay API after user approves payment
 * @see {@link
https://developers.google.com/pay/api/web/reference/object#PaymentData|PaymentData object
reference}
 */
function processPayment(paymentData) {
    // show returned data in developer console for debugging
    // console.log(paymentData);
    // @todo pass payment token to your gateway to process payment
    paymentToken = paymentData.paymentMethodData.tokenizationData.token;

    var promise = sendPaymentData(paymentToken);
    promise.then(function(response) {
        if (response.error) {
            // console.log(response.error);
        } else {
        }
    });
}
}

```

В ответ Google должен вернуть элемент PaymentData, а поле paymentMethodData.tokenizationData.token должно содержать надежно зашифрованный токен Google Pay (строка символов). Пример Google Pay Token:

```
{  
    "signature":  
        "MEUCIQDY3wBQyHB4sZcktRoJXKxm+OLcjHzCvdDeGn23oX0kkwIgKznRFZZL+sDMv1b5cuD+YurXMZraYBsr9hbr  
        avVY5Ro\u003d",  
    "protocolVersion": "ECv2",  
    "signedMessage":  
        "{\"encryptedMessage\": \"cI87tLqzqTGyCFnMMCVWcTHw3xhYIK+CEnuQ74K+n1LpCgOlfpScib9jds4sxDtN  
        6CunCqCSMfd/3yHeeRy6aCx1yyqcT4ey6NueeBznprJpkmVVgi1JHWLQt4hzAXMUAcYASYLOabKP9fUZvHkOBDytD  
        531jpzNXa+Spc/zrpGzFKx2C4VU9sC95q9i+ey+kr7ZMNVCOFJPWXu71KZ105IOOqozJ6/70MKmxP3jM89eeq+/19  
        QnyHjQLXfnQPvQjiUJKGCcRKDLLrb3XoY5ZUUzGfN5eZCLzCVg0hWEbwU+6J7KWYJyW+Wrlr8bagN9zWsrMKhDpsQ  
        bHfyzb+yBzFUoxeUgL4a7FeVvE11IcHtqsvTCf6FENV20aF5VLDv5qzUkV+PzTAibFEuabA0God9UbVCVVv7nM8QF  
        zvRPhzYYFVFTn4JHvL2qZ4pAR91E+w\\u003d\\u003d\", \"ephemeralPublicKey\": \"BPHLC4sBHpenY1M0i  
        xmiDMuWJTatJOqggRUwtgBJMcBp28VsxD7zPI7985x4F5EjMP5y8j/cuUzbe/cGPjOKGk\\u003d\", \"tag\": \"  
        \"RaXrPOUuc5iw3oxDa0C2MOjaKxgxIRQvwOspmtFV0zU\\u003d\""}  
}
```

Проведение платежа

Процедура аналогична проведению платежа при совершении оплаты из мобильного приложения (см. выше).

Методы API: платежи

Отправка запросов, если не указано иное, осуществляется методом POST с заголовком «Content-Type: application/x-www-form-urlencoded».

Регистр символов и порядок следования параметров в запросах не важен.

Регистр символов в url-адресах запросов — важен.

Ответ на запрос передается в формате JSON, кодировка UTF-8.

StartSession

Запрос инициализации платежа. Выполняется перед перенаправлением пользователя на страницу платежного шлюза Payler.

URL запроса: <https://{{host}}.payler.com/gapi/StartSession>

Доступно для: gapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
type	A	R	Тип операции. Определяет количество стадий платежа. OneStep — одностадийный платеж; TwoStep — двухстадийный платеж
session_type	N1	O	Тип сессии. Определяет режим сохранения карточных данных. Если этот параметр равен 1, требуется передавать customer_id 0 — обычная страница оплаты; 1 — страница оплаты с возможностью выбора из сохраненных карт или ввода полных карточных данных для оплаты картой не из списка. При выборе сохраненной карты пользователю необходимо ввести только CVV. По умолчанию — 0
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Для каждого платежа (сессии) требуется использовать уникальный идентификатор. Только печатные ASCII-символы
customer_id	A..100	O	Идентификатор пользователя. Определяет пользователя, чей список карт будет доступен для выбора (если session_type = 1), или за которым будет сохранена карта, используемая при оплате. Должен быть передан, если параметр session_type = 1. Пользователь создается автоматически, если данное поле пустое

Название	Тип	R/O	Описание
currency	A3	O	Валюта платежа (RUB, USD, EUR, GBP, PLN, TJS, KGS). По умолчанию — RUB
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.)
product	A..256	O	Описание заказа, платежа
template	A..100	O	Используемый шаблон страницы оплаты, заданный мерчантом. При отсутствии используется шаблон «по умолчанию».
lang	A2	O	Предпочитаемый язык платёжной формы и ответов сервера. en — английский; ru — русский. По умолчанию — ru
userdata	A..1000	O	Пользовательские данные. Можно передать любую информацию, которую нужно сохранить вместе с платежом, а потом получить её с помощью метода GetAdvancedStatus
recurrent	N1	O	Показывает, требуется ли создать шаблон рекуррентных платежей на основе текущего: 1 — требуется создать шаблон; 0 — шаблон создавать не требуется
pay_page_param_*	A..100	O	Параметры для отображения на странице оплаты. Можно передать любые параметры, начинающиеся с pay_page_param_, а потом отобразить их на странице оплаты. Для этого необходимо использовать шаблон страницы (см. параметр template)
pay_page_param_addtoreport3	A	O	При сплитовании : суммы зачисления для каждого получателя в минимальной денежной единице (копейки, центы и т. д.) через запятую
pay_page_param_addtoreport4	A	O	При сплитовании : ID получателей через запятую. Регистрируются через службу технической поддержки
antifraud_*	A..100	O	Дополнительные параметры для системы антифрод. Можно передать любые параметры, начинающиеся с antifraud_
payment_methods	A..500	O	Способы платежа для данной сессии. Если параметр не указан, то будут доступны все способы, разрешенные мерчанту. Через запятую перечислены способы платежа, доступные для сессии. См. Способы оплаты
email	A..100	R	Адрес электронной почты пользователя
return_url_success	A..1000	O	URL, на который будет перенаправлен

Название	Тип	R/O	Описание
			пользователь в случае успешного прохождения платежа
return_url_decline	A..1000	O	URL, на который будет перенаправлен пользователь в случае неуспешного прохождения платежа
lifetime	N	O	Время жизни сессии в минутах, от 1 мин и более

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Соответствует переданному в запросе
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.)
session_id	A..100	R	Идентификатор платёжной сессии

Пример ответа на успешный запрос:

```
{
  "amount": 30000,
  "session_id": "FE6nrdcfw4Zy88CRki6sjc1mRxW9xcv7m7CS",
  "order_id": "d1434908-7260-483e-8254-fa43af1b835d"
}
```

PayGate

Запрос списания средств с перенаправлением пользователя на страницу шлюза. Выполняется после команды [StartSession](#). Результатом обработки запроса является списание денежных средств при одностадийной схеме проведения платежа, либо блокировка средств на карте пользователя при двухстадийной схеме проведения платежа.

URL запроса: [https://\[host\].payler.com/gapi/Pay](https://[host].payler.com/gapi/Pay)

Доступно для: gapi

Метод отправки запроса: GET

Параметры запроса:

Название	Тип	R/O	Описание
session_id	A..100	R	Идентификатор платёжной сессии. Содержится в ответе на запрос StartSession

После обработки запроса PayGate на стороне платежного шлюза пользователь будет перенаправлен на страницу с результатами платежа, а через 3 секунды возвращен на сайт мерчанта.

Адрес возврата пользователя (URL страницы) — указывается мерчантом заранее в виде URL. Пример адреса возврата: http://myshop.ru/complete?order_id={order_id}

Для получения результатов транзакции следует использовать данные, полученные в рамках запроса статуса транзакции [GetStatus](#).

Нужно учесть, что возврат на указанный URL может быть сделан несколько раз, например, когда пользователь запутался, нажимал кнопку Back в браузере и несколько раз вводил карточные данные для оплаты. Если пользователь ошибочно попытается повторно оплатить заказ в рамках одной сессии, то ему будет показано сообщение о том, что оплата уже произведена, и сделано перенаправление на сайт мерчанта.

FindSession

Поиск платёжной сессии по идентификатору платежа (*order_id*).

URL запроса: <https://{host}.payler.com/{api}/FindSession>

Доступно для: gapi, mapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор оплачиваемого заказа или перевода в системе мерчанта. Только печатные ASCII-символы

Параметры ответа:

Название	Тип	R/O	Описание
id	A..100	R	Идентификатор платежной сессии
created	A	R	Время создания сессии в формате «уууу-ММ-дд HH:mm:ss»
valid_through	A	R	Время действия сессии в формате «уууу-ММ-дд HH:mm:ss»
type	A	R	Тип операции. Определяет количество стадий платежа. OneStep — одностадийный платеж; TwoStep — двухстадийный платеж
customer_id	A..100	O	Идентификатор пользователя
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Только печатные ASCII-символы

Название	Тип	R/O	Описание
amount	N	R	Сумма платежа в минимальной денежной единице (копейки, центы и т. д.)
product	A..256	R	Наименование продукта.
currency	A..3	R	Валюта платежа (RUB, USD, EUR, GBP, PLN, TJS, KGS). По умолчанию — RUB
pay_page_params	D	R	Параметры для отображения на странице оплаты
userdata	A..10000	R	Пользовательские данные
lang	A2	R	Предпочитаемый язык платёжной формы и ответов сервера. en — английский, ru — русский
recurrent	B	R	Показывает, требуется ли создать шаблон рекуррентных платежей: 1 — шаблон создан, 0 — шаблон не требуется
session_type	A	R	Тип сессии. Определяет режим сохранения карточных данных. Если этот параметр равен 1, требуется передавать customer_id 0 — обычная страница оплаты; 1 — страница оплаты с возможностью выбора из сохраненных карт или ввода полных карточных данных для оплаты картой не из списка. При выборе сохраненной карты пользователю необходимо ввести только CVV

Пример ответа на успешный запрос (платёж):

```
{
  "id": "VLaFQpI88NpCncTA1TkhlX6HtkhwQAKhxvz",
  "created": "2015-10-26 17:11:30",
  "valid_through": "2015-10-26 17:11:30",
  "type": "OneStep",
  "order_id": "ad7ad8b4-d50e-4b68-72f4-ca1264a8fae4",
  "amount": 30000,
  "product": "el-ticket",
  "currency": "RUB",
  "pay_page_params": {"pay_page_param_phone": "+790012345678",
  "pay_page_param_city": "Москва"},
  "userdata": "data",
  "lang": "RU",
  "recurrent": true,
  "customer_id": "kfRWk8nbT9LhO19sVYotkJ2gHSPs5E4Qph35",
  "session_type": "0"
}
```

PayMerchant

Запрос выполняется в рамках одностадийной схемы проведения платежа.
Результат обработки запроса — списание денежных средств с карты пользователя.

URL запроса: <https://{{host}}.payler.com/mapi/v1/Pay>

Доступно для: mapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Для каждого платежа нужен свой уникальный идентификатор. Только печатные ASCII-символы
currency	A3	O	Валюта платежа (RUB, USD, EUR, GBP, PLN, TJS, KGS). По умолчанию — RUB
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.)
card_number	A..19	O	Номер карты. Десятичные цифры без разделителей [0–9]
card_holder	A..100	O	Имя владельца карты. Латинские символы, десятичные цифры или символ пробела [a-zA-Z0-9]
expired_year	N2	O	Год окончания срока действия карты
expired_month	N2	O	Месяц окончания срока действия карты
secure_code	A	R	Код подлинности карты (CVC2/CVV2). Десятичные цифры [0–9]
lang	A2	O	Предпочитаемый язык ответов сервера. en — английский; ru — русский. По умолчанию — ru
email	A..100	R	Адрес электронной почты пользователя
userdata	A..1000	O	Пользовательские данные. Можно передать в этой строке любую информацию, которую нужно сохранить вместе с платежом, а потом получить её с помощью метода GetAdvancedStatus
recurrent	N1	O	Показывает, требуется ли создать шаблон рекуррентных платежей на основе текущего. 1 — требуется создать шаблон; 0 — шаблон создавать не требуется

Название	Тип	R/O	Описание
card_id	A..100	O	Идентификатор сохранённой карты. Если указан, то можно не указывать card_number, expired_year, expired_month, достаточно указать secure_code
save_card	B	O	Показывает, требуется ли сохранить данные карты. Если передан этот параметр, то нужно передавать также и customer_id. 1 — требуется сохранить карту; 0 — карту сохранять не требуется
customer_id	A..100	O	Идентификатор пользователя. Если передан параметр save_card равный true или 1, то показывает, для какого пользователя будет сохранена карта. Если передан параметр card_id, то показывает, какому пользователю принадлежит карта (значение должно совпадать с указанным при сохранении карты). Генерируется мерчантом
antifraud_*	A..100	O	Дополнительные параметры для системы антифрод. Можно передать любые параметры, начинающиеся с antifraud_
user_entered_*	A..100	O	Можно указать дополнительные поля для оплаты, начинающиеся с user_entered_. Затем все эти поля возвращаются в GetAdvancedStatus в поле user_entered_params
payer_ip	A	R	IP-адрес браузера покупателя
browserAccept	A	R	Заголовок Accept из браузера покупателя
browserLanguage	A	R	Язык браузера устройства покупателя. en — английский; ru — русский
browserUserAgent	A	R	Содержание заголовка User-Agent в браузере покупателя
browserJavaEnabled	B	R	Признак возможности выполнения JavaScript в браузере устройства покупателя
browserScreenHeight	A	R	Общая высота экрана устройства покупателя в пикселях
browserScreenWidth	A	R	Общая ширина экрана устройства покупателя в пикселях
browserColorDepth	A	R	Глубина цветопередачи в битах
browserTZ	A	R	Часовой пояс — разница (в минутах) во времени между временем по UTC и местным временем браузера на устройстве покупателя

Название	Тип	R/O	Описание
threeDsNotificationUrl	A	R	Url перенаправления после выполнения ChallengeComplete
pay_page_param_addtoreport3	A	O	При сплитовании : суммы зачисления для каждого получателя в минимальной денежной единице (копейки, центы и т. д.) через запятую
pay_page_param_addtoreport4	A	O	При сплитовании : ID получателей через запятую. Регистрируются через службу технической поддержки

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Соответствует переданному в запросе
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.). Соответствует переданной в запросе
auth_type	N1	R	Значение, указывающее, требуется ли дополнительная авторизация для осуществления платежа. 0 — дополнительная авторизация не требуется; 1 — авторизация с использованием 3D-Secure
recurrent_template_id	A..100	O	Идентификатор шаблона рекуррентных платежей. Присутствует, если в рамках текущей транзакции был создан шаблон рекуррентных платежей
card_id	A..100	O	Идентификатор сохранённой карты
card_status	A	O	Статус карты. Saved — карта сохранена и готова к использованию. Invalid — карта недействительна (например, истёк срок её действия)
card_number	A..19	R	Маскированный номер банковской карты, к которой привязан шаблон. Указывается при совершении первого платежа в серии рекуррентных платежей. Десятичные цифры без разделителей [0–9] и маскировочный символ ‘x’
card_holder	A..100	R	Имя держателя карты
expired_month	N2	R	Месяц истечения срока действия карты
expired_year	N2	R	Год истечения срока действия карты
acs_url	A..*	O	URL для перенаправления пользователя для прохождения 3DS (Access Control Service)

Название	Тип	R/O	Описание
md	A*	O	Номер транзакции в МПС для идентификации при прохождении 3DS 1.0 (Merchant Data)
pareq	A*	O	Данные по транзакции для прохождения 3DS 1.0 (Payment Authentication Request)
threeDS_server_transID	A	O	Уникальный идентификатор, присвоенный 3DS-сервером. Необходим вызов 3DSMethod
threeDS_method_url	A	O	Значение для передачи в запросе к 3DSMethod
creq	A	O	Данные для авторизации в 3DS 2.0 после прохождения Challenge
status	A..10	O	Статус операции

Примеры ответов на успешный запрос:

Если карта вовлечена в 3DS 2.0 и требуется вызов [3DSMethod](#):

```
{
    "threeDS_server_transID": "9f7f012d-d6d0-38fe-9bfd-70a727c7473a",
    "threeDS_method_url": "https://someurl.com",
    "auth_type": 1,
    "amount": 2050,
    "order_id": "c8ebb763-5b75-4c5e-a731-2489220b4f2a"
}
```

Если карта вовлечена в 3DS 2.0 и требуется вызов [ChallengeComplete](#):

```
{
    "acs_url": "https://someurl.com",
    "creq": "eyJ0aHJ1ZURTU2VydmVyVHJhbnnJRCI6ImE3ZTYzZWRkLTywYTgtMzE0Yi04MzJ1LTU3YTFhNzg2NTZjZSIsImFjc1RyYW5zSUQiOiJhM2MxMDQwYS05OGUxLTQ0N2EtOTE4Ni0zYzY3N2MyODMxNmEiLCJjaGFsbGVz2VXaW5kb3dTaXplIjoiMDUiLCJtZXNzYWdlVHlwZSI6IkNSZXElCJtZXNzYWdlVmVyc2lvbiI6IjIuMS4wIn0=",
    "auth_type": 1,
    "amount": 2050,
    "card_number": "411111xxxxxx1111",
    "card_holder": "NONAME",
    "expired_year": 20,
    "expired_month": 12,
    "order_id": "201d6446-c351-4df6-8f46-8f5b24be76aa"
}
```

Если карта вовлечена в 3DS 1.0:

```
{
    "acs_url": "https://web.rbsuat.com/acs/auth/start.do",
    "pareq": "eJxVUdluwjAQ/BWU9+ADhwBajNICALUBWkK19i3HF1KRAxMq4OtrByhU8sPMejRjz8LgmG0bP6j2aZH3LdakVgPzUEjSfN23VsHY71gDCCFGI06XGB8USvBxxw/X2EiTvvXluOi6HWFjRIUTusyxIx4Lm1FkPBKUJu3Ikrdw3nAn4Ro kdU6TA71R7ajiTZhXEsJ49zSdScHdNqVArhQyVNOhZLwlnLYl5EihDzOUZXjaogJSE4iLQ16pk2yLFpAbgYPayk1 V1t1CLvJmXGRAzBjIPXxxMGivbY5pIv3g9eyf13wejNjs2xezs8f84cizD70+EKOAJKxQcsq61OWtBu30uD4MSD2 HMDP50qk/csfQmgjv4eJxALpfpeu/Pf/GAI91kaNW6NL+MJD7e58nprq40q2UbizoNfpUbjUOJi/55/sqGe/Ws6W HH6bQWmQcU90M67KLpSFAjA257opc16zRv/X/Aj0Rt1Q=",
    "md": "949cdebb-b7a7-46b0-874e-bd11f410992c",
    "auth_type": 1,
    "amount": 500,
    "card_number": "411111xxxxxx1111",
    "card_holder": "IVAN",
    "expired_year": 19,
    "expired_month": 12,
    "order_id": "test_example_002"
}
```

Если карта не вовлечена в 3DS или аутентификация успешно прошла по frictionless flow и выполнение [3DSMethod](#) не требовалось:

```
{
    "auth_type": 0,
    "amount": 500,
    "recurrent_template_id": "rec-pay-42301e79-b92d-4753-b248-b92ad552fdfd",
    "card_number": "555555xxxxxx5599",
    "card_holder": "IVAN",
    "expired_year": 19,
    "expired_month": 12,
    "order_id": "test_example_003",
    "status": "Charged"
}
```

Внимание

Если в запросе в качестве значения параметра recurrent было указано true, транзакция была выполнена успешно, но в ответе не вернулся идентификатор шаблона, то транзакция прошла по обычной схеме, без создания шаблона рекуррентных платежей.

После совершения запроса PayMerchant требуется вызвать запрос [GetStatus](#) для получения актуального статуса операции.

Карты Maestro

Правила МПС запрещают проводить рекуррентные платежи по картам Maestro. Мы можем настроить ваш аккаунт так, чтобы при попытке оплаты платежной сессии с созданием рекуррентного платежа (в запросе в качестве значения параметра recurrent указано true) платеж отклонялся, а пользователь получал сообщение, что данный тип карт не поддерживается.

Block

Запрос выполняется в рамках двухстадийной схемы проведения платежа.
Результат обработки запроса — блокировка денежных средств на карте пользователя.

URL запроса: <https://{{host}}.payler.com/mapi/v1/Block>

Доступно для: mapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Для каждого платежа нужен свой уникальный идентификатор. Только печатные ASCII-символы
currency	A3	O	Валюта платежа (RUB, USD, EUR, GBP, PLN, TJS, KGS). По умолчанию — RUB
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.)
card_number	A..19	O	Номер карты. Десятичные цифры без разделителей [0–9]
card_holder	A..100	O	Имя владельца карты. Латинские символы, десятичные цифры или символ пробела [a-zA-Z0-9]
expired_year	N2	O	Год окончания срока действия карты
expired_month	N2	O	Месяц окончания срока действия карты. Целое двухзначное число в формате с лидирующим нулем
secure_code	A	R	Код подлинности карты (CVC2/CVV2). Десятичные цифры [0–9]
email	A..100	R	Адрес электронной почты пользователя
lang	A2	O	Предпочитаемый язык ответов сервера. en — английский; ru — русский. По умолчанию — ru
userdata	A..*	O	Пользовательские данные. Можно передать в этой строке любую информацию, которую нужно сохранить вместе с платежом, а потом получить её с помощью метода GetAdvancedStatus

Название	Тип	R/O	Описание
recurrent	A..100	O	Показывает, требуется ли создать шаблон рекуррентных платежей на основе текущего. 1 — требуется создать шаблон; 0 — шаблон создавать не требуется
card_id	A..100	O	Идентификатор ранее сохранённой карты. Если указан, то можно не указывать card_number, expired_year, expired_month, достаточно указать secure_code
save_card	N1	O	Показывает, требуется ли сохранить данные карты. Если передано 1, то нужно передавать customer_id 1 — требуется сохранить карту; 0 — карту сохранять не требуется
customer_id	A..100	O	Идентификатор пользователя. Если передан параметр save_card, равный true или 1, то показывает для какого пользователя будет сохранена карта. Если передан параметр card_id, то показывает, какому пользователю принадлежит карта (значение должно совпадать с указанным при сохранении карты). Генерируется мерчантом
user_entered_*	A..100	O	Можно указать дополнительные поля для оплаты, начинающиеся с user_entered_. Затем все эти поля возвращаются в GetAdvancedStatus в поле user_entered_params
pay_page_param_addtoreport3	A	O	При сплитовании : суммы зачисления для каждого получателя в минимальной денежной единице (копейки, центы и т. д.) через запятую
pay_page_param_addtoreport4	A	O	При сплитовании : ID получателей через запятую. Регистрируются через службу технической поддержки
antifraud_*	A..100	O	Дополнительные параметры для системы антифрод. Можно передать любые параметры, начинающиеся с antifraud_
payer_ip	A	R	IP-адрес браузера покупателя
browserAccept	A	R	Заголовок Accept из браузера покупателя
browserLanguage	A	R	Язык браузера устройства покупателя. en — английский; ru — русский
browserUserAgent	A	R	Содержание заголовка User-Agent в браузере покупателя

Название	Тип	R/O	Описание
browserJavaEnabled	B	R	Признак возможности выполнения JavaScript в браузере устройства покупателя
browserScreenHeight	A	R	Общая высота экрана устройства покупателя в пикселях
browserScreenWidth	A	R	Общая ширина экрана устройства покупателя в пикселях
browserColorDepth	A	R	Глубина цветопередачи в битах
browserTZ	A	R	Часовой пояс — разница (в минутах) во времени между временем по UTC и местным временем браузера на устройстве покупателя
threeDsNotificationUrl	A	R	Url перенаправления после выполнения ChallengeComplete

Параметры ответа:

Формат ответа совпадает форматом ответа [PayMerchant](#)

Внимание

Если в запросе в качестве значения параметра recurrent было указано true, транзакция была выполнена успешно, но в ответе не вернулся идентификатор шаблона, то транзакция прошла по обычной схеме, без создания шаблона рекуррентных платежей.

3DSMethod

Данный метод вызывается на стороне банка-эмитента карты, если в ответе [PayMerchant](#), [Block](#) или [PaySaveCard](#) вернулись значения *threeDS_server_transID* и *threeDS_method_url*. Мерчанту необходимо добавить на страницу пользователя javascript — iframe с html-формой с *threeDSmethodData*. Javascript выполняет submit этой формы, выполняя POST-запрос на *threeDSmethodUrl* (см. значение *threeDS_method_url* в ответе PayMerchant, Block или PaySaveCard).

Пример содержимого iframe:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>3DS Method</title>
    <script type="text/javascript"
        language="javascript">function moveWindow() { document.returnform.submit(); } /script>
</head>
<body onLoad="javascript:moveWindow()">
    <form name="returnform" action=' + threeDsMethodUrl + '" method="POST">
        <input type="hidden" name="threeDSMethodData" value=' + threeDSMethodData + '" />
        <noscript>
            <center>Please click the submit button below.<br /><input type="submit" name="submit"
value="Submit" />
        </center>
        </noscript>
    </form>
</body>
</html>
```

Сценарий выполнения:

1. Система мерчанта в скрытом iframe отправляет методом POST на ThreeDSMethodURL параметр *threeDSmethodData*. На данном шаге *threeDSmethodData* — это закодированный в base64url объект json с полями:
 - **threeDSMethodNotificationURL** — адрес, на который ACS отправит POST запрос после того, как соберет информацию о браузере клиента;
 - **threeDSServerTransID** — значение параметра *threeDS_method_url*.
2. ACS отправляет страницу со скриптом для сбора информации о браузере на стороне пользователя;
3. ACS собирает информацию о браузере пользователя;
4. Собранная информация о браузере отправляется на ACS;
5. ACS отправляет POST запрос на адрес *threeDSMethodNotificationURL*. В запросе передает параметр *threeDSmethodData*. На данном шаге *threeDSmethodData* — это закодированный в base64url объект json с полем *threeDSServerTransID*.

После получения от ACS POST-запроса на адрес *threeDSMethodNotificationURL* или, если в течение предусмотренного протоколом таймаута (10 сек) запрос не был получен, система мерчанта выполняет запрос [ThreeDsMethodComplete](#).

ThreeDsMethodComplete

Завершение аутентификации по протоколу 3DS 2.0. Выполняется после вызова [3DSMethod](#).

Формат запроса: <https://{{host}}.payler.com/{{api}}/v1/ThreeDsMethodComplete>

Доступно для: gapi, mapi

Параметры запроса:

Название	Тип	R/O	Описание
threeDs_comp_ind	A1	R	Результат прохождения аутентификации 3DS 2.0. Возможные значения: Y — успех (callback от ACS был получен), N — отклонение (callback от ACS не был получен)
key	A	R	Платёжный ключ мерчанта
threeDS_server_transID	A	R	Значение threeDSServerTransID, полученное в ответе на запрос PayMerchant , Block или PaySaveCard

Параметры ответа:

Название	Тип	R/O	Описание
acs_url	A	O	URL перенаправления пользователя для прохождения 3DS
creq	A	O	Данные для авторизации в 3DS 2.0 после Challenge
auth_type	N1	R	Значение, указывающее, требуется ли дополнительная авторизации для осуществления платежа. 0 — дополнительная авторизация не требуется; 1 — авторизация с использованием 3D-Secure
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.). Соответствует переданной в запросе
recurrent_template_id	A..100	O	Идентификатор шаблона рекуррентных платежей. Присутствует, если в рамках текущей транзакции был создан шаблон рекуррентных платежей
card_number	A..19	R	Маскированный номер банковской карты, к которой привязан шаблон. Указывается при совершении первого платежа в серии рекуррентных платежей. Десятичные цифры без разделителей [0–9] и маскировочный символ ‘x’
card_holder	A..100	R	Имя держателя карты
expired_year	N2	R	Год истечения срока действия карты
expired_month	N2	R	Месяц истечения срока действия карты
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Соответствует переданному в запросе
status	A..10	O	Статус операции

Примеры ответов на успешный запрос:

Если карта вовлечена в 3DS 2.0 и требуется вызов [ChallengeComplete](#):

```
{
    "acs_url": "https://someurl.com",
    "creq": "eyJ0aHJ1ZURTU2VydmVyVHJhbNJRCl6ImE3ZTYzzWRkLTywYTgtMzE0Yi
04MzJ1LTU3YTFhNzg2NTZjZSIsImFjc1RyYW5zSUQiOiJhM2MxMDQwYS05OGUxLTQ0
N2EtOTE4Ni0zYzY3N2MyODMxNmEiLCJjaGFsbGVuZ2VXaW5kb3dTaXplIjoiMDUiLC
JtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lvbiI6IjIuMS4wIn0=",
    "auth_type": 1,
    "amount": 2050,
    "card_number": "411111xxxxxx1111",
    "card_holder": "NONAME",
    "expired_year": 20,
    "expired_month": 12,
    "order_id": "201d6446-c351-4df6-8f46-8f5b24be76aa"
}
```

Если карта не вовлечена в 3DS или аутентификация успешно прошла по frictionless flow:

```
{
    "auth_type": 0,
    "amount": 500,
    "recurrent_template_id": "rec-pay-42301e79-b92d-4753-b248-
b92ad552fdfd",
    "card_number": "555555xxxxxx5599",
    "card_holder": "IVAN",
    "expired_year": 19,
    "expired_month": 12,
    "order_id": "test_example_003",
    "status": "Charged"
}
```

ChallengeComplete

Прохождение дополнительного этапа для проверки подлинности платежа 3D-Secure 2.0 при блокировке или списании средств со счёта карты.

Сценарий:

- Если требуется дополнительный этап выполнения проверки подлинности платежа 3D-Secure 2.0, то в ответе метода [PayMerchant](#), [Block](#), [PaySaveCard](#) или [ThreeDsMethodComplete](#) присутствуют строковые параметры `acs_url` и `creq`.
 - После получения ответа от шлюза мерчант перенаправляет пользователя на сайт банка-эмитента для дополнительной аутентификации.
- Формат перенаправления:** POST-запрос по адресу, указанному в значении параметра `acs_url`. **Параметр:** `creq`.

Формат ответа в результате перенаправления: возврат пользователя POST-запросом по

адресу, указанному в параметре threeDsNotificationUrl запроса [PayMerchant](#), [Block](#), [PaySaveCard](#) или [ThreeDsMethodComplete](#). Параметр: cres.

Формат запроса: <https://{{host}}.payler.com/mapi/v1/ChallengeComplete>

Доступно для: мари

Параметры запроса:

Название	Тип	R/O	Описание
cres	A	R	Значение seq, полученное при ответе на запрос PayMerchant , Block , PaySaveCard или ThreeDsMethodComplete
key	A	R	Платёжный ключ мерчанта

Параметры ответа:

Формат ответа совпадает форматом ответа [Send3DS](#)

Send3DS

Прохождение аутентификации 3D-Secure 1.0 при блокировке или списании средств с карты.

Сценарий:

- Если карта пользователя задействована в механизме аутентификации 3D-Secure, значение параметра *auth_type* в ответе на запрос [PayMerchant](#), [Block](#) или [PaySaveCard](#) становится равным *ThreeDS* (1). В этом случае в ответе также присутствуют строковые параметры *acs_url*, *pareq* и *md*.
- После получения ответа от шлюза мерчант перенаправляет пользователя на сайт банка-эмитента для дополнительной аутентификации.
Формат перенаправления: POST-запрос по адресу, указанному в значении параметра *acs_url*. **Параметры:** *PaReq*, *TermUrl*, *MD*.
Формат ответа в результате перенаправления:
Возврат пользователя POST-запросом по адресу, указанному в значении атрибута *TermUrl*. **Параметры:** *PaRes*, *MD*.
- После получения результатов аутентификации от банка-эмитента для завершения блокировки или платежа мерчант выполняет дополнительный запрос к шлюзу Payler.

Формат запроса: <https://{{host}}.payler.com/mapi/v1/Send3DS>

Доступно для: мари

Параметры запроса:

Название	Тип	R/O	Описание
PaRes	A	R	Значение, полученное страницей <i>TermUrl</i> от банка-эмитента
MD	A	R	Значение, полученное при ответе на запрос PayMerchant , Block или PaySaveCard

Параметры ответа:

Название	Тип	R/O	Описание
auth_type	N1	R	Значение, указывающее, требуется ли дополнительная авторизация для осуществления платежа. 0 — дополнительная авторизация не требуется; 1 — авторизация с использованием 3D-Secure
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.). Соответствует переданной в запросе
recurrent_template_id	A..100	O	Идентификатор шаблона рекуррентных платежей. Присутствует, если в рамках текущей транзакции был создан шаблон рекуррентных платежей
card_number	A..19	R	Маскированный номер банковской карты, к которой привязан шаблон. Указывается при совершении первого платежа в серии рекуррентных платежей. Десятичные цифры без разделителей [0–9] и маскировочный символ ‘x’
card_holder	A..100	R	Имя держателя карты
expired_year	N2	R	Год истечения срока действия карты
expired_month	N2	R	Месяц истечения срока действия карты
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Соответствует переданному в запросе
status	A..10	O	Статус операции

Пример ответа на успешный запрос:

```
{
    "auth_type": 0,
    "amount": 500,
    "recurrent_template_id": "rec-pay-42301e79-b92d-4753-b248-b92ad552fdf",
    "card_number": "555555xxxxxx5599",
    "card_holder": "IVAN",
    "expired_year": 19,
    "expired_month": 12,
    "order_id": "test_example_003",
    "status": "Charged"
}
```

Charge

Запрос выполняется для списания предварительно заблокированной суммы с карты пользователя.

Запрос выполняется после успешной команды [PayGate](#) или [Block](#) при двухстадийной схеме проведения платежа. Статус платежа должен быть «Authorized». Также, для схемы Gate, в запросе [StartSession](#), параметр *type* должен иметь значение «TwoStep».

URL запроса: <https://{{host}}.payler.com/{{api}}/Charge>

Доступно для: gapi, mapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
password	A..100	R	Пароль мерчанта для проведения операций. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор заказа в системе мерчанта. Должен соответствовать order_id операции StartSession или Block
amount	N	R	Сумма списания в минимальной денежной единице (копейки, центы и т. д.). Должна точно совпадать с суммой заблокированных средств

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Соответствует переданному в запросе
amount	N	R	Списанная сумма в минимальной денежной единице (копейки, центы и т. д.)
status	A..10	R	Статус операции

Пример ответа на успешный запрос:

```
{
    "amount": 3457500,
    "order_id": "sk-d97b9c351818d820fdd3d5d33e6c1edf",
    "status": "Charged"
}
```

Retrieve

Запрос выполняется после успешной команды [PayGate](#) или [Block](#) при двухстадийной схеме проведения платежа. Статус платежа должен быть «Authorized». Результатом обработки запроса является разблокировка (частичная или полная) денежных средств на карте пользователя.

URL запроса: <https://{{host}}.payler.com/{{api}}/Retrieve>

Доступно для: gapi, mapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
password	A..100	R	Пароль мерчанта для проведения операций. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор заказа в системе мерчанта. Должен соответствовать order_id операции StartSession или Block
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.), которую следует вернуть. Не должна превышать значение, указанное в запросе StartSession или PayMerchant

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Соответствует переданному в запросе
new_amount	N	R	Новая величина суммы платежа в копейках

Пример ответа на успешный запрос:

```
{
    "new_amount": 10000,
    "order_id": "4c1552c5-11bb-465f-8564-62ac933ae15d"
}
```

Refund

Запрос выполняется после успешной команды [PayGate](#) или [PayMerchant](#) при одностадийной схеме, либо после успешной команды Charge при двухстадийной схеме.
Статус платежа должен быть «Charged». Результатом обработки запроса является возврат (полный или частичный) списанных денежных средств на карту пользователя.

Формат запроса: <https://{{host}}.payler.com/{{api}}/Refund>

Доступно для: gapi, mapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
password	A..100	R	Пароль мерчанта для проведения операций. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор заказа в системе мерчанта. Должен соответствовать order_id операции StartSession , PayMerchant или Block
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.), которую следует вернуть

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Соответствует переданному в запросе
amount	N	R	Остаток списанной суммы в минимальной денежной единице (копейки, центы и т. д.)

Пример ответа на успешный запрос:

```
{
    "amount": 0,
    "order_id": "4c1552c5-11bb-465f-8564-62ac933ae15d"
}
```

RepeatPay

Запрос осуществления повторного платежа в рамках серии рекуррентных платежей. Результатом обработки запроса является списание денежных средств без указания пользователем карточных данных.

Формат запроса: <https://{{host}}.payler.com/{{api}}/v1/RepeatPay>

Доступно для: gapi, mapr

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Для каждого платежа (сессии) требуется использовать уникальный идентификатор. Только печатные ASCII-символы
amount	N	R	Сумма платежа в минимальной денежной единице (копейки, центы и т. д.). Может отличаться от суммы платежа, осуществленного в рамках транзакции, на основе которой был создан шаблон рекуррентных платежей
recurrent_template_id	A..100	O	Идентификатор шаблона рекуррентных платежей, полученный в callback
currency	A3	O	Валюта платежа (RUB, USD, EUR, GBP, PLN, TJS, KGS). По умолчанию — RUB
userdata	A..1000	O	Пользовательские данные. Можно передать в этой строке любую информацию, которую нужно сохранить вместе с платежом, а потом получить её с помощью метода GetAdvancedStatus

Параметр currency

Если мерчант использует при расчетах более одной валюты, параметр currency является обязательным!

Например, если шаблон рекуррентного платежа был создан по платежу, осуществленному в USD, то при повторном списании необходимо указывать USD.

В противном случае платеж пройдет по рублевому терминалу, с конвертацией валюты и начислением дополнительной комиссии.

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор оплаченного заказа в системе мерчанта. Соответствует переданному в запросе
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.)
status	A..10	R	Статус операции

Пример ответа на успешный запрос:

```
{
    "amount": 50000,
    "order_id": "3e31f52f-84bd-4a98-b798-8aaf3d325a229",
    "status": "Charged"
}
```

Карты Maestro

Правила МПС запрещают проводить рекуррентные платежи по картам Maestro. Мы можем настроить аккаунт в Payler так, чтобы при попытке оплаты платежной сессии с созданием рекуррентного платежа (в запросе [StartSession](#) в качестве значения параметра *recurrent* было указано *true*) платеж отклонялся, а пользователь получил сообщение, что данный тип карт не поддерживается.

GetTemplate

Запрос получения информации о шаблоне рекуррентных платежей. Рекомендуется использовать для получения полной информации о зарегистрированном шаблоне рекуррентных платежей.

Формат запроса: <https://{{host}}.payler.com/{{api}}/GetTemplate>

Доступно для: gapi, mapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
recurrent_template_id	A..100	R	Идентификатор шаблона рекуррентных платежей, полученный в callback

Параметры ответа:

Название	Тип	R/O	Описание
recurrent_template_id	A..100	R	Идентификатор шаблона рекуррентных платежей. Соответствует переданному в запросе
created	A	R	Дата и время регистрации шаблона рекуррентных платежей в формате «уууу-ММ-дд HH:mm:ss»
card_holder	A..100	R	Имя держателя карты, к которой привязан шаблон. Указывается при совершении первого платежа в серии рекуррентных платежей
card_number	A..19	R	Маскированный номер банковской карты, к которой

Название	Тип	R/O	Описание
			привязан шаблон. Указывается при совершении первого платежа в серии рекуррентных платежей. Десятичные цифры без разделителей [0–9] и маскировочный символ 'x'
expiry	A..5	R	Срок действия шаблона рекуррентных платежей. Месяц и год в формате «MM/yy»
active	B	R	Показывает, активен ли шаблон. 1 — шаблон активен; 0 — шаблон неактивен

Пример ответа на успешный запрос:

```
{
    "recurrent_template_id": "rec-pay-2160e16c-849d-42c5-a224-19f302ce4806",
    "created": "2014-08-21 18:22:03",
    "card_holder": "TEST USER",
    "card_number": "510047xxxxxx0401",
    "expiry": "03/15",
    "active": true
}
```

ActivateTemplate

Запрос активации/деактивации шаблона рекуррентных платежей.

Используется для временного отключения возможности совершать платежи по шаблону.

Формат запроса: [https://\[host\].payler.com/{api}/ActivateTemplate](https://[host].payler.com/{api}/ActivateTemplate)

Доступно для: gapi, mapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
recurrent_template_id	A..100	R	Идентификатор шаблона рекуррентных платежей
active	B	R	Показывает, требуется ли активировать или деактивировать шаблон рекуррентных платежей. 1 — шаблон требуется активировать; 0 — шаблон требуется деактивировать

Параметры ответа:

Название	Тип	R/O	Описание
recurrent_template_id	A..100	R	Идентификатор шаблона рекуррентных платежей.

Название	Тип	R/O	Описание
			Соответствует переданному в запросе
created	A	R	Дата и время регистрации шаблона рекуррентных платежей в формате «yyyy-MM-dd HH:mm:ss»
card_holder	A..100	R	Имя держателя карты, к которой привязан шаблон. Указывается при совершении первого платежа в серии рекуррентных платежей
card_number	A..19	R	Маскированный номер банковской карты, к которой привязан шаблон. Указывается при совершении первого платежа в серии рекуррентных платежей. Десятичные цифры без разделителей [0–9] и маскировочный символ 'x'
expiry	A..5	R	Срок действия шаблона рекуррентных платежей в формате «MM/yy»
active	B	R	Показывает, активен ли шаблон. 1 — шаблон активен; 0 — шаблон неактивен

Пример ответа на успешный запрос:

```
{
    "recurrent_template_id": "rec-pay-2160e16c-849d-42c5-a224-19f302ce4806",
    "created": "2014-08-21 18:22:03",
    "card_holder": "TEST USER",
    "card_number": "510047xxxxxx0401",
    "expiry": "03/15",
    "active": true
}
```

GetStatus

Результатом выполнения запроса является получение актуального статуса операции. Рекомендуется использовать в случае неполучения ответа от шлюза Payler при проведении других запросов по платежу.

Формат запроса: <https://{{host}}.payler.com/{{api}}/GetStatus>

Доступно для: gapi, mapi, cgapi, cmapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор операции в системе мерчанта

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор платежа в системе мерчанта. Соответствует переданному в запросе
amount	N	R	Сумма платежа в минимальной денежной единице (копейки, центы и т. д.)
status	A..10	R	Состояние платежа. См. Статусы операций
recurrent_template_id	A..100	O	Идентификатор шаблона рекуррентных платежей. Присутствует, если в рамках текущей транзакции был создан шаблон рекуррентных платежей или она была осуществлена по шаблону
payment_type	A	O	Способ оплаты. См. Способы оплаты

Пример ответа на успешный запрос:

```
{
    "status": "Charged",
    "amount": 30000,
    "recurrent_template_id": "rec-pay-02e20707-eed4-4cb4-9a45-03b5465f8e92",
    "order_id": "a0c799a6-a3df-47cc-b452-63b19d95bb59",
    "payment_type": "Card"
}
```

Внимание

Если статус операции имеет значение «Pending», то запрос статуса следует повторять до изменения статуса. Частоту запроса изменять в геометрической прогрессии.

Если в запросе [StartSession](#), [Block](#) или [PayMerchant](#) в качестве значения параметра recurrent было указано true, транзакция была выполнена успешно, но в ответе на запрос [GetStatus](#) не вернулся идентификатор шаблона, то транзакция прошла по обычной схеме, без создания шаблона рекуррентных платежей.

GetAdvancedStatus

Результатом выполнения запроса является получение расширенного статуса операции. Используется для получения детальной информации в том случае, если ответа на запрос [GetStatus](#) недостаточно для решения бизнес-задач.

Формат запроса: <https://{{host}}.payler.com/{{api}}/GetAdvancedStatus>

Доступно для: gapi, mapi, cgapi, smapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор операции в системе мерчанта

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор платежа в системе мерчанта. Соответствует переданному в запросе
amount	N	R	Сумма платежа в минимальной денежной единице (копейки, центы и т. д.)
status	A..10	R	Состояние платежа. См. Статусы операций
product	A..256	O	Описание операции
recurrent_template_id	A..100	O	Идентификатор шаблона рекуррентных платежей. Присутствует, если в рамках текущей транзакции был создан шаблон рекуррентных платежей или она была осуществлена по шаблону
card_number	A..19	R	Маскированный номер карты, с которой сделан платеж
card_holder	A..100	R	Имя держателя карты, с которой осуществлен платеж
dt	A	R	Время регистрации транзакции в системе Payler в формате «уууу-ММ-дд HH:mm:ss»
from	A	O	IP-адрес, с которого выполнен запрос платежа
approval_code	A..6	O	Буквенно-цифровой код, присвоенный банком-эмитентом для удостоверения подтверждения запроса авторизации
rrn	A..6	O	Номер, присвоенный транзакции в платежной системе
userdata	A..1000	O	Значение параметра userdata, переданного в методе StartSession , PayMerchant или Block
type	A..7	O	Определяет количество стадий платежа. OneStep — одностадийный платеж; TwoStep — двухстадийный платеж
processing	A	O	Название процессинга, обрабатывающего платёж
processing_order_id	A	O	Идентификатор платежа в процессинговом центре
card_bankname	A	R	Название банка. Может быть пустым значением
card_paymentsystem	A	R	Платежная система. Может быть пустым значением
card_product	A	R	Тип карты. Может быть пустым значением

Название	Тип	R/O	Описание
card_id	A..100	O	Идентификатор карты, с помощью которой была произведена оплата. Отсутствует, если оплата производилась не по сохранённой карте
card_status	A..10	O	Статус карты: Saved — карта сохранена и готова к использованию; Invalid — карта недействительна (например, истёк срок её действия)
user_entered_params	D	R	При оплате можно указать дополнительные поля для ввода, начинающиеся с user_entered_. Максимальная длина каждого параметра — 100 символов. Затем все эти поля возвращаются в GetAdvancedStatus в поле user_entered_params

Пример ответа на успешный запрос (платёж):

```
{
    "card_number": "554781xxxxxx4672",
    "card_holder": "NO NAME",
    "expired_year": 22,
    "expired_month": 3,
    "dt": "2019-06-26 12:59:05",
    "from": "127.0.0.1",
    "approval_code": "598101",
    "rrn": "917791485893",
    "processing": "PSB",
    "currency": "RUB",
    "user_entered_params": {
        "user_entered_documentSerial": "1500",
        "user_entered_documentNumbers": "aB"
    },
    "type": "OneStep",
    "card_bankname": "PUBLIC JOINT STOCK COMPANY PROMSVYAZBANK",
    "card_paymentsystem": "MASTERCARD",
    "card_product": "GOLD",
    "card_country_code": "RU",
    "processing_order_id": "1561543145757000",
    "status": "Charged",
    "amount": 500,
    "recurrent_template_id": "rec-pay-da24a6e3-8b9e-4bf2-8fa5-45e5365f5222",
    "payment_type": "Card",
    "order_id": "Dr24ake3-875e-4bf2-8fa5-45e5365f6773"
}
```

Внимание

Если в запросе StartSession, Block или PayMerchant в качестве значения параметра recurrent было указано true, транзакция была выполнена успешно, но в ответе на запрос GetAdvancedStatus не вернулся идентификатор шаблона, то транзакция прошла по обычной схеме без создания шаблона рекуррентных платежей.

ApplePay

Проведение платежа с помощью Apple Pay через мобильное приложение или на web-сайте мерчанта. Результатом обработки запроса является списание или блокировка денежных средств на карте пользователя.

URL запроса: <https://{host}.payler.com/{api}/ApplePay>

Content-Type: application/json

Доступно для: gapi, mapr

Параметры запроса:

Название	Тип	R/O	Описание
session_id	A..100	R	Идентификатор сессии, полученный с помощью вызова StartSession
apple_payment_data	A	R	Зашифрованные платежные данные из параметра paymentData объекта PKPaymentToken
email	A..100	R	Адрес электронной почты плательщика
apple_pay_type	N1	R	Тип оплаты. Возможные значения: 1 — оплата из приложения, 3 — кнопка Apple Pay на сайте мерчанта
payler_ip	A	O	IP-адрес пользователя

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Соответствует переданному в запросе StartSession
amount	N	R	Сумма платежа в минимальной денежной единице (копейки, центы и т. д.)

Пример ответа на успешный запрос:

```
{
    "amount": 30000,
    "order_id": "d1434908-7260-483e-8254-fa43af1b835d",
}
```

AppleValidateMerchant

Валидация мерчанта в Apple.

URL запроса: <https://{{host}}.payler.com/{{gapi}}/applevalidatemerchant>

Content-Type: application/json

Доступно для: gapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A	R	Платежный ключ мерчанта
apple_pay_validation_url	A	R	Url валидации мерчанта, полученный обработчиком события onvalidatemerchant
apple_pay_type	N1	R	Тип оплаты. 3 — кнопка Apple Pay на сайте мерчанта

Параметры ответа:

Название	Тип	R/O	Описание
validation_object	A	R	Объект, полученный от Apple. Его необходимо передать в метод completeMerchantValidation .

Пример ответа на успешный запрос:

```
{
  "validation_object": "..."
}
```

GooglePay

Проведение платежа с помощью Google Pay через мобильное приложение или на web-сайте мерчанта. Результатом обработки запроса является списание или блокировка денежных средств на карте пользователя.

Для использования метода GooglePay мерчанту требуется предварительно выполнить настройку Google Pay API. Информацию по настройке [см. выше](#).

URL запроса: <https://{{host}}.payler.com/{{api}}/GooglePay>

Content-Type: application/json

Доступно для: gapi, mapr

Параметры запроса:

Название	Тип	R/O	Описание
session_id	A..100	R	Идентификатор сессии полученный с помощью вызова StartSession
google_pay_token	A	R	Зашифрованные платежные данные "paymentMethodData", "tokenizationData", "token" из полученного от Google объекта PaymentData (можно получить в методе handlePaymentSuccess) вызовом paymentData, getPaymentMethodToken, getToken
email	A..100	O	Адрес электронной почты плательщика
payer_ip	A	O	IP-адрес пользователя

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Соответствует переданному в запросе StartSession
amount	N	R	Сумма платежа в минимальной денежной единице (копейки, центы и т. д.)

Пример ответа на успешный запрос:

```
{
    "amount": 30000,
    "order_id": "d1434908-7260-483e-8254-fa43af1b835d",
}
```

Методы API: переводы

Отправка запросов, если не указано иное, осуществляется методом POST с заголовком «Content-Type: application/x-www-form-urlencoded».

Регистр символов и порядок следования параметров в запросах не важен.

Регистр **символов в url-адресах запросов** — важен.

Ответ на запрос передается в формате JSON, кодировка UTF-8.

StartCreditSession

Запрос инициализации перевода. Выполняется перед перенаправлением пользователя на страницу платежного шлюза Payler.

URL запроса: [https://\[host\].payler.com/cqapi/StartSession](https://[host].payler.com/cqapi/StartSession)

Доступно для: cgapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
password	A..100	R	Пароль мерчанта для проведения операций. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор перевода в системе мерчанта. Для каждого перевода (сессии) требуется использовать уникальный идентификатор. Только печатные ASCII-символы
session_type	N1	O	Тип сессии относительно сохранения карточных данных. При сохранении карточных данных последующие переводы можно совершать просто выбором нужной карты. Если этот параметр равен 1, то нужно передавать также и customer_id. 0 — обычная страница перевода, без сохранения карточных данных, 1 — показать пользователю список сохранённых карт и возможность выбора из списка или ввода полных карточных данных для перевода на карту не из списка. По умолчанию — 0
customer_id	A..100	O	Идентификатор пользователя. Если параметр session_type равен 1, то на странице перевода будет возможность выбрать ранее сохранённую карту (для данного пользователя) или ввести новую. Генерируется мерчантом
amount	N	R	Сумма перевода в минимальной денежной единице (копейки, центы и т. д.)

Название	Тип	R/O	Описание
product	A..256	O	Описание перевода
template	A..100	O	Используемый шаблон страницы перевода, заданный мерчантом. При отсутствии используется шаблон «по умолчанию»
lang	A2	O	Предпочитаемый язык платёжной формы и ответов сервера. en — английский; ru — русский. По умолчанию — ru
userdata	A..1000	O	Пользовательские данные. Можно передать в этой строке любую информацию, которую нужно сохранить вместе с переводом, а потом получить её с помощью метода GetAdvancedStatus
email	A..100	O	Адрес электронной почты пользователя
pay_page_param_*	A..100	O	Параметры для отображения на странице перевода. Можно передать любые параметры, начинающиеся с pay_page_param_, а потом отобразить их на странице оплаты. Для этого необходимо использовать шаблон страницы (см. параметр template)
pay_page_param_addtoreport3	A	O	При сплитовании : суммы переводов для каждого получателя в минимальной денежной единице (копейки, центы и т. д.) через запятую
pay_page_param_addtoreport4	A	O	При сплитовании : ID получателей через запятую. Регистрируются через службу технической поддержки

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор перевода в системе мерчанта. Соответствует переданному в запросе
amount	N	R	Сумма перевода в минимальной денежной единице (копейки, центы и т. д.).
session_id	A..100	R	Идентификатор платёжной сессии

Пример ответа на успешный запрос:

```
{
    "amount": 30000,
    "session_id": "b088175c-eca6-4317-aa84-74de5080ca20",
    "order_id": "d1434908-7260-483e-8254-fa43af1b835d"
}
```

CreditGate

Перенаправление пользователя с сайта мерчанта на страницу формы оплаты на стороне шлюза. Выполняется после команды [StartCreditSession](#). После заполнения пользователем полей формы, шлюз автоматически выполняет запрос к эмитенту.

Результатом обработки запроса является перечисление денежных средств на карту пользователя. После обработки запроса на стороне платежного шлюза пользователь будет перенаправлен на страницу с результатами операции, а через 3 секунды возвращен на сайт мерчанта.

URL запроса: [https://\[host\].payler.com/cqapi/Credit](https://[host].payler.com/cqapi/Credit)

Доступно для: csgapi

Метод отправки запроса: GET

Параметры запроса:

Название	Тип	R/O	Описание
session_id	A..100	R	Идентификатор платёжной сессии. Содержится в ответе на запрос StartCreditSession

После обработки запроса [CreditGate](#) на стороне платежного шлюза пользователь будет перенаправлен на страницу с результатами перевода, а через 3 секунды будет возвращен на сайт мерчанта.

Адрес возврата пользователя (URL страницы) — указывается мерчантом заранее в виде URL. Пример адреса возврата: http://myshop.ru/complete?order_id={order_id}

Для получения результатов транзакции следует использовать данные, полученные в рамках запроса статуса транзакции (см. метод [GetStatus](#)).

Необходимо учесть, что возврат на указанный URL может быть сделан несколько раз, например когда пользователь запутался, нажимал кнопку «Back» в браузере и несколько раз вводил свои карточные данные для перевода. Если пользователь ошибочно попытается повторно осуществить перевод в рамках одной сессии, то ему будет показано сообщение о том, что перевод уже произведен и сделано перенаправление на сайт мерчанта.

CreditMerchant

Запрос перевода денежных средств на карту пользователя.

Формат запроса: <https://{{host}}.payler.com/cmapi/Credit>

Доступно для: смари

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
password	A..100	R	Пароль мерчанта для проведения операций. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор перевода в системе мерчанта. Для каждого перевода нужен свой уникальный идентификатор. Только печатные ASCII-символы
amount	N	R	Сумма перевода в минимальной денежной единице (копейки, центы и т. д.)
currency	A3	O	Валюта платежа (RUB, USD, EUR, GBP, PLN, TJS, KGS). По умолчанию — RUB
card_number	A..19	R	Номер карты. Десятичные цифры без разделителей [0–9]
card_holder	A..100	O	Имя держателя карты. Латинские символы, десятичные цифры, символ пробела [a-zA-Z0-9]
lang	A2	O	Предпочитаемый язык ответов сервера. en — английский; ru — русский. По умолчанию — ru
email	A..100	R	Адрес электронной почты пользователя
userdata	A..100 0	O	Пользовательские данные. Можно передать в этой строке любую информацию, которую нужно сохранить вместе с переводом, а потом получить её с помощью метода GetAdvancedStatus
user_entered_params	A..100	O	Можно указать дополнительные поля для перевода, начинающиеся с user_entered_. Затем все эти поля возвращаются в GetAdvancedStatus в поле user_entered_params
pay_page_param_addtoreport3	A	O	При сплитовании : суммы переводов для каждого получателя в минимальной денежной единице (копейки, центы и т. д.) через запятую
pay_page_param_addtoreport4	A	O	При сплитовании : ID получателей через запятую. Регистрируются через службу технической поддержки

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерчанта. Соответствует переданному в запросе
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.)
card_holder	A..100	R	Имя держателя карты. Соответствует переданному в запросе
card_number	A..19	O	Маскированный номер карты, с помощью которой осуществлен перевод
status	A..10	R	Статус перевода

Пример ответа на успешный запрос:

```
{
    "amount": 100,
    "card_number": "555555xxxxxx5599",
    "card_holder": "test",
    "order_id": "19",
    "status": "Pending"
}
```

После совершения запроса [CreditMerchant](#) требуется вызвать запрос [GetStatus](#) для получения актуального статуса операции.

RepeatCredit

Запрос перевода средств на заранее сохраненную карту. Результатом обработки запроса является перевод денежных средств.

URL запроса: [https://\[host\].payler.com/cqapi/RepeatCredit](https://[host].payler.com/cqapi/RepeatCredit)

Доступно для: cqapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
password	A..100	R	Пароль мерчанта для проведения операций через Credit Gate API. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор перевода в системе мерчанта. Для каждого перевода нужен свой уникальный идентификатор. Только печатные ASCII-символы
customer_id	A..100	O	Идентификатор пользователя. Генерируется мерчантом
amount	N	R	Сумма перевода в минимальной денежной единице (копейки, центы и т. д.)

Название	Тип	R/O	Описание
currency	A3	O	Валюта платежа (RUB, USD, EUR, GBP, PLN, TJS, KGS). По умолчанию — RUB
card_id	A..100	R	Идентификатор карты на которую будет осуществлен перевод
lang	A2	O	Предпочитаемый язык ответов сервера. en — английский; ru — русский. По умолчанию — ru
userdata	A..1000	O	Пользовательские данные. Можно передать в этой строке любую информацию, которую нужно сохранить вместе с платежом, а потом получить её с помощью метода GetAdvancedStatus

Параметр currency

Если мерчант использует при расчетах более одной валюты, параметр currency является обязательным!

Например, если шаблон рекуррентного платежа был создан по переводу, осуществленному в USD, то при повторном переводе необходимо указывать USD.

В противном случае операция пройдет по рублевому терминалу, с конвертацией валюты и начислением дополнительной комиссии.

Параметры ответа:

Название	Тип	R/O	Описание
order_id	A..100	R	Идентификатор перевода в системе мерчанта. Соответствует переданному в запросе
amount	N	R	Сумма в минимальной денежной единице (копейки, центы и т. д.)
card_holder	A..100	O	Имя держателя карты. Соответствует переданному в запросе
card_number	A..19	R	Маскированный номер карты, с помощью которой осуществлен перевод
status	A..10	R	Статус перевода

Пример ответа на успешный запрос:

```
{
    "amount": 100,
    "card_number": "555555xxxxxx5599",
    "card_holder": "test",
    "order_id": "19",
    "status": "Pending"
}
```

Если статус перевода имеет значение «Pending», то запрос статуса следует повторять до изменения статуса. Частоту запроса изменять в геометрической прогрессии. После совершения запроса RepeatCredit требуется вызвать запрос GetStatus для получения актуального статуса операции.

GetBalance

Запрос списка балансов.

URL запроса: [https://\[host\].payler.com/{api}/v1/GetBalance](https://[host].payler.com/{api}/v1/GetBalance)

Доступно для: cгари, смари

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа

Параметры ответа:

Название	Тип	R/O	Описание
balance	Ar	R	Баланс в минимальных единицах валюты
name	A	R	Наименование счета
currency	A3	R	Код валюты
remain	N	R	Остаток

Пример ответа на успешный запрос:

```
{
    "balance": [
        {
            "name": "Card (Euro)",
            "currency": "EUR",
            "remain": 0
        },
        {
            "name": "Card (United States Dollar)",
            "currency": "USD",
            "remain": 0
        },
        {
            "name": "Card (Russian Ruble)",
            "currency": "RUB",
            "remain": 30000
        },
        {
            "name": "Qiwi (Russian Ruble)",
            "currency": "RUB",
            "remain": 7000
        }
    ]
}
```

Методы API: фискальные чеки

Receipt

Запрос выдачи чека.

URL запроса: <https://{{host}}.payler.com/kkt/v2/Receipt>

Content-Type: application/json

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
type	A	R	Тип операции, которая должна быть выполнена. Возможные значения: <ul style="list-style-type: none"> • sell — чек «Приход», • sell_refund — чек «Возврат прихода», • sell_correction — чек «Коррекция прихода», • buy — чек «Расход», • buy_refund — чек «Возврат расхода», • buy_correction — чек «Коррекция расхода»
order_id	A..100	R	Идентификатор заказа в системе мерчанта. Соответствует значению, переданному при создании платежа
items	Ar..100	R	Продаваемые товары. Ограничение по количеству — от 1 до 100. Список полей в каждом товаре см. ниже
email	A..100	O	Адрес электронной почты пользователя. Должен присутствовать по крайней мере один из параметров email или phone
phone	A..15	O	Номер телефона пользователя. Должен присутствовать по крайней мере один из параметров — email или phone
client_name	A..256	O	Наименование покупателя (клиента)
client_inn	N12	O	ИНН покупателя (клиента)
sno	A	O	Система налогообложения. <ul style="list-style-type: none"> • osn — общая СН, • usn_income — упрощенная СН (доходы), • usn_income_outcome — упрощенная СН (доходы минус расходы), • envd — единый налог на вмененный доход, • esn — единый сельскохозяйственный налог, • patent — патентная СН
payments	Ar..10	O	Оплаты. Список полей в каждом товаре см. ниже
vats	Ar	O	Атрибуты налогов на чек. Ограничение по количеству от 1 до 6.

Название	Тип	R/O	Описание
			Необходимо передать либо сумму налога на позицию, либо сумму налога на чек. Если будут переданы и сумма налога на позицию и сумма налога на чек, сервис учтет только сумму налога на чек. Список полей см. ниже
supplier_info	Ar	O	Телефоны поставщика. См. описание ниже
agent_type	A	O	<p>Признак агента по предмету расчёта (ограничен агентами, введенными в ККТ при фискализации).</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> • bank_paying_agent — банковский платежный агент. Оказание услуг покупателю (клиенту) лицом, являющимся банковским платежным агентом; • bank_paying_subagent — банковский платежный субагент. Оказание услуг покупателю (клиенту) лицом, являющимся банковским платежным субагентом; • paying_agent — платежный агент. Оказание услуг покупателю (клиенту) лицом, являющимся платежным агентом; • paying_subagent — платежный субагент. Оказание услуг покупателю (клиенту) лицом, являющимся платежным субагентом; • attorney — поверенный. Осуществление расчета с покупателем (клиентом) лицом, являющимся поверенным; • commission_agent — комиссионер. Осуществление расчета с покупателем (клиентом) лицом, являющимся комиссионером; • another — другой тип агента. Осуществление расчета с покупателем (клиентом) лицом, являющимся агентом и не являющимся банковским платежным агентом (субагентом), платежным агентом (субагентом), поверенным, комиссионером
cashier	A..64	O	Ф. И. О. кассира

Поля товаров (массив items):

Название	Тип	R/O	Описание
name	A..256	R	Наименование товара
price	F	R	Цена в рублях. Целая часть — не более 8 знаков; дробная часть — не более 2 знаков
quantity	F	R	Количество/вес. Целая часть — не более 8 знаков; дробная часть — не более 3 знаков
sum	F	R	Сумма позиции в рублях. Целая часть — не более 8 знаков; дробная часть — не более 2 знаков. Если значение sum меньше значения (price*quantity), то разница является скидкой на позицию

Название	Тип	R/O	Описание
measurement_unit	A..16	O	Единица измерения товара, работы, услуги, платежа, выплаты, иного предмета расчета
nomenclature_code	A	O	Код товара в шестнадцатеричном представлении с пробелами. Максимальная длина – 32 байта. Пример: 00 00 00 01 00 21 FA 41 00 23 05 41 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 12 00 AB 00
payment_method	A	R	Признак способа расчета. Возможные значения: <ul style="list-style-type: none"> • full_prepayment — предоплата 100%. Полная оплата до момента передачи предмета расчета; • prepayment — предоплата. Частичная предварительная оплата до момента передачи предмета расчета; • advance — аванс; • full_payment — полный расчет. Полная оплата, в том числе с учетом аванса; • partial_payment — частичный расчет и кредит. Частичная оплата предмета расчета в момент его передачи с последующей оплатой в кредит; • credit — передача в кредит. Передача предмета расчета без его оплаты в момент его передачи с последующей оплатой в кредит; • credit_payment — оплата кредита. Оплата предмета расчета после его передачи с оплатой в кредит
payment_object	A	R	Признак предмета расчета. Возможные значения: <ul style="list-style-type: none"> • commodity — товар. О реализуемом товаре, за исключением подакцизного товара (наименование и иные сведения, описывающие товар); • excise — подакцизный товар. О реализуемом подакцизном товаре (наименование и иные сведения, описывающие товар); • job — работа. О выполняемой работе (наименование и иные сведения, описывающие работу); • service — услуга. Об оказываемой услуге (наименование и иные сведения, описывающие услугу); • gambling_bet — ставка азартной игры. О приеме ставок при осуществлении деятельности по проведению азартных игр; • gambling_prize — выигрыш азартной игры. О выплате денежных средств в виде выигрыша при осуществлении деятельности по проведению азартных игр; • lottery — лотерейный билет. О приеме средств при реализации лотерейных билетов, электронных лотерейных билетов, приеме ставок при осуществлении деятельности по проведению лотерей; • lottery_prize — выигрыш лотереи. О выплате денежных средств в виде выигрыша при осуществлении деятельности по проведению лотерей; • intellectual_activity — предоставление результатов интеллектуальной деятельности. О предоставлении прав на использование результатов интеллектуальной

Название	Тип	R/O	Описание
			<p>деятельности или средств индивидуализации;</p> <ul style="list-style-type: none"> • payment — платеж. Об авансе, задатке, предоплате, кредите, взносе в счет оплаты, пени, штрафе, вознаграждении, бонусе и ином аналогичном предмете расчета; • agent_commission — агентское вознаграждение. О вознаграждении пользователя, являющегося платежным агентом (субагентом), банковским платежным агентом (субагентом), комиссионером, поверенным или иным агентом; • property_right — имущественное право. О передаче имущественных прав; • non-operating_gain — внереализационный доход. О внереализационном доходе; • insurance_premium — страховые взносы. О суммах расходов, уменьшающих сумму налога (авансовых платежей) в соответствии с пунктом 3.1 статьи 346.21 Налогового кодекса Российской Федерации; • sales_tax — торговый сбор. О суммах уплаченного торгового сбора; • resort_fee — курортный сбор. О курортном сборе; • composite — составной предмет расчета. О предмете расчета, состоящем из предметов, каждому из которых может быть присвоено значение вышеупомянутых признаков; • another — иной предмет расчета. О предмете расчета, не относящемся к вышеупомянутым предметам расчета
vat	Ar	R	Атрибуты налога на позицию. Необходимо передать либо сумму налога на позицию, либо сумму налога на чек. Если будут переданы и сумма налога на позицию и сумма налога на чек, сервис учета только сумму налога на чек. Список полей см. ниже
supplier_info	Ar	O	Данные поставщика. Список полей см. ниже

Оплаты (массив payments):

Название	Тип	R/O	Описание
type	N	R	<p>Вид оплаты. Возможные значения:</p> <p>1 — безналичный; 2 — предварительная оплата (зачет аванса и (или) предыдущих платежей); 3 — постоплата (кредит); 4 — иная форма оплаты (встречное предоставление); 5–9 — расширенные виды оплаты. Для каждого фискального типа оплаты можно указать расширенный вид оплаты</p>
sum	F	R	Сумма к оплате в рублях. Целая часть не более 8 знаков, дробная часть не более 2 знаков

Налоги (массив `vats` и объект `items.vat`):

Название	Тип	R/O	Описание
sum	F	R/O	Сумма налога в рублях. <i>Необязателен в объекте items.vat. В массиве vats обязательен.</i> Целая часть — не более 8 знаков; дробная часть — не более 2 знаков
type	A	R	Устанавливает номер налога в ККТ. Возможные значения: <ul style="list-style-type: none">• none — без НДС;• vat0 — НДС по ставке 0%;• vat10 — НДС чека по ставке 10%;• vat110 — НДС чека по расчетной ставке 10/110;• vat20 — НДС чека по ставке 20%;• vat120 — НДС чека по расчетной ставке 20/120

Поставщик (массив `supplier_info` в массиве `items`):

Название	Тип	R/O	Описание
phones	Ar	O	Номера телефонов поставщика. Каждый элемент массива — А..19. Номер необходимо передать вместе с кодом страны без пробелов и дополнительных символов, кроме символа «+». Если номер начинается с символа «+», то максимальная длина одного элемента массива — 19 символов. Если номер относится к РФ (префикс «+7»), то значение можно передать без префикса (номер «+7 925 1234567» можно передать как «9251234567»). Максимальная длина одного элемента массива в таком случае – 17 символов
name	A	O	Наименование поставщика
inn	A	O	ИНН поставщика

Поставщик (массив `supplier_info`):

Название	Тип	R/O	Описание
phones	Ar	O	Соответствует описанию <code>supplier_phones</code> в массиве <code>items</code> (см. выше)

Пример запроса:

```
{
    "key": "d37232ca-cf1d-42f2-a4c3-fb59690e581a",
    "order_id": "0e0c0d63-9e29-4d32-a2a3-3c94fb93dace",
    "client_name": "some name",
    "client_inn": "123456",
    "items": [
        {
            "name": "колбаса Клинский Брауншвейгская с/к в/с",
            "vat": 10,
            "type": "vat10"
        }
    ]
}
```

```

    "price": 1000.00,
    "quantity": 0.6,
    "sum": 600.00,
    "measurement_unit": "кг",
    "payment_method": "full_payment",
    "payment_object": "commodity",
    "nomenclature_code": "00 00 00 01 00 21 FA 41 00 23 05 41 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 12 00 AB 00",
    "vat": {
        "type": "vat20"
    },
    "supplier_info": {
        "phones": [
            "8-111-111-11-11"
        ],
        "name": "ООО «Лютик»",
        "inn": "956839506500"
    }
},
"payments": [
{
    "type": 1,
    "sum": 500.0
}
],
"vats": [
{
    "type": "vat20",
    "sum": 45.76
}
],
"type": "sell",
"email": "test@example.com",
"phone": "9268448477",
"sno": "osn",
"supplier_info": {
    "phones": [
        "8-111-111-11-11"
    ]
},
"agent_type": "commission_agent",
"cashier": "Голева А. С."
}
}

```

Параметры ответа:

В ответ возвращается идентификатор чека.

Название	Тип	R/O	Описание
receipt_id	A	R	Идентификатор чека

Пример ответа:

```
{
    "receipt_id": "123"
}
```

GetReceiptStatus

Запрос получения информации по чекам. В данном запросе возвращается массив со всеми чеками, зарегистрированными в рамках одного заказа.

Формат запроса: [https://\[host\].payler.com/kkt/v2/GetStatus](https://[host].payler.com/kkt/v2/GetStatus)

Content-Type: application/json

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор заказа в системе мерчанта. Соответствует значению, переданному при создании платежа

Пример запроса:

```
{
    "key": "d37232ca-cf1d-42f2-a4c1-fb59690e581a",
    "order_id": "0e0c0d63-9e29-4d32-a2a3-3c94fb93dace"
}
```

Параметры ответа:

В ответ возвращается массив Receipts с информацией по каждому чеку, выданному по данной сессии.

Поля каждого объекта в массиве:

Название	Тип	R/O	Описание
receipt_id	A	R	Идентификатор чека. Равен возвращённому в ответ на запрос Receipt
items	Ar..100	R	Продаваемые товары. Список полей в каждом товаре см. выше. В случае создания чека через /kkt/v2/Receipt поля payment_method, payment_object и vat будут присутствовать в ответе. В случае создания чека через /kkt/v1/Receipt поля tax, tax_sum будут присутствовать в ответе
email	A..100	O	Адрес электронной почты пользователя
phone	A..100	O	Номер телефона пользователя
uuid	A..100	O	Уникальный идентификатор
status	A	R	Статус обработки чека: <ul style="list-style-type: none"> • created — создан, • wait — ожидание, • done — готово, • fail — ошибка

Название	Тип	R/O	Описание
fn_number	A	O	Номер ФН
shift_number	N	O	Номер смены
receipt_date_time	A	O	Дата и время документа из ФН в формате «yyyy-MM-dd HH:mm:ss»
fiscal_receipt_number	N	O	Номер чека в смене
fiscal_document_number	N	O	Фискальный номер документа
ecr_registration_number	A	O	Регистрационный номер ККТ
fiscal_document_attribute	N	O	Фискальный признак документа
error_code	N	O	Код ошибки. Присутствует, если произошла ошибка при обработке чека
error_text	A	O	Текст ошибки. Присутствует, если произошла ошибка при обработке чека
error_type	A	O	Тип ошибки. Присутствует, если произошла ошибка при обработке чека
vats	Ar	O	Атрибуты налогов на чек
client_name	A..256	O	Наименование покупателя (клиента)
client_inn	N12	O	ИНН покупателя (клиента)
payments	Ar..10	O	Оплаты
agent_type	A	O	Признак агента по предмету расчёта
supplier_info	Ar	O	Данные поставщика. Список полей см. выше
sno	A	O	Система налогообложения. <ul style="list-style-type: none"> • osn — общая СН, • usn_income — упрощенная СН (доходы), • usn_income_outcome — упрощенная СН (доходы минус расходы), • envd — единый налог на вмененный доход, • esn — единый сельскохозяйственный налог, • patent — патентная СН
cashier	A...64	O	Ф. И. О. кассира

Пример ответа:

Методы API: сохранение данных карт

Отправка запросов осуществляется методом POST с заголовком «Content-Type: application/x-www-form-urlencoded».

Регистр символов и порядок следования параметров **в запросах** не важен.

Регистр символов **в url-адресах запросов** — важен.

Ответ на запрос передается в формате JSON, кодировка UTF-8

StartSaveCardSession

Запрос инициализации сессии сохранения карты. Выполняется перед перенаправлением пользователя на страницу платежного шлюза Payler.

Формат запроса: [https://\[host\].payler.com/\[api\]/StartSaveCardSession](https://[host].payler.com/[api]/StartSaveCardSession)

Доступность: gapi, cgapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
customer_id	A..100	R	Идентификатор плательщика. Возвращается в ответ на запрос CustomerRegister
template	A..100	O	Используемый шаблон страницы оплаты, заданный мерчантом. При отсутствии используется шаблон «по умолчанию»
lang	A2	O	Предпочитаемый язык платёжной формы и ответов сервера. en — английский; ru — русский. По умолчанию — ru
currency	A3	O	Валюта платежа (RUB, USD, EUR, GBP, PLN, TJS, KGS). По умолчанию — RUB
pay_page_param_*	A..100	O	Параметры для отображения на странице оплаты. Можно передать любые параметры, начинающиеся с pay_page_param_, а потом отобразить их на странице оплаты. Для этого необходимо использовать шаблон страницы (см. параметр template)
return_url_success	A..1000	O	URL, на который будет перенаправлен пользователь в случае успешного сохранения карты
lifetime	N	O	Время жизни сессии в минутах, от 1 минуты и более

Параметры ответа:

Название	Тип	R/O	Описание
session_id	A..100	R	Идентификатор сессии
order_id	A..100	R	Идентификатор заказа в системе мерчанта. Генерируется автоматически

Пример ответа на успешный запрос:

```
{
    "session_id": "dOadCwvzuHxJVKNX4peVZpzTRu1VG4OJwomj",
    "order_id": "kqonTcrezjZHhNxs619HYoVE737N7JNXYadb"
}
```

Save

Перенаправление пользователя с сайта мерчанта на страницу формы оплаты на стороне шлюза. Выполняется после команды StartSaveCardSession. После заполнения пользователем полей формы, шлюз автоматически выполняет запрос к эмитенту. Результатом является сохранение данных карты пользователя

Формат запроса: <https://{{host}}.payler.com/{{api}}/Save>

Доступно для: gapi, cgapi

Метод отправки запроса: GET

Параметры запроса:

Название	Тип	R/O	Описание
session_id	A..100	R	Идентификатор платёжной сессии. Содержится в ответе на запрос StartSaveCardSession

После обработки запроса Save на стороне платежного шлюза пользователь будет перенаправлен на страницу с отображением результата сохранения карты, а через 3 секунды будет возвращен на сайт мерчанта.

Адрес возврата пользователя (URL страницы) — указывается мерчантом заранее в виде URL. Пример адреса возврата: http://mysite.ru/complete?session_id={{session_id}}

Для получения результатов операции следует использовать данные, полученные в рамках запроса статуса операции (см. метод [GetStatusSaveCard](#)).

Нужно учесть, что возврат на указанный URL может быть сделан несколько раз, например, когда пользователь запутался, нажимал кнопку «Back» в браузере. Если пользователь ошибочно попытается повторно сохранить карту в рамках одной сессии, то ему будет показано сообщение о том, что карта уже сохранена и сделано перенаправление на сайт мерчанта.

PaySaveCard

Сохранение данных карты путем списания или блокировки (в зависимости от настроек мерченшта) контрольной суммы на счете карты с последующей автоматической разблокировкой или возвратом списанной суммы.

Результатом является сохранение данных карты пользователя.

URL запроса: [https://\[host\].payler.com/mapi/v1/PaySaveCard](https://[host].payler.com/mapi/v1/PaySaveCard)

Доступно для: mapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерченшта. Выдается с параметрами доступа
order_id	A..100	R	Идентификатор оплачиваемого заказа в системе мерченшта. Для каждого платежа нужен свой уникальный идентификатор. Только печатные ASCII-символы
currency	A3	O	Валюта платежа (RUB, USD, EUR, GBP, PLN, TJS, KGS). По умолчанию — RUB
card_number	A..19	R	Номер карты. Десятичные цифры без разделителей [0–9]
card_holder	A..100	O	Имя владельца карты. Латинские символы, десятичные цифры или символ пробела [a-zA-Z0-9]
expired_year	N2	R	Год окончания срока действия карты
expired_month	N2	R	Месяц окончания срока действия карты
secure_code	A	R	Код подлинности карты (CVC2/CVV2). Десятичные цифры [0–9]
lang	A2	O	Предпочитаемый язык ответов сервера. en — английский; ru — русский. По умолчанию — ru
email	A..100	R	Адрес электронной почты пользователя
userdata	A..1000	O	Пользовательские данные. Можно передать в этой строке любую информацию, которую нужно сохранить вместе с платежом, а потом получить её с помощью метода GetAdvancedStatus
customer_id	A..100	O	Идентификатор пользователя. Показывает для какого пользователя будет сохранена карта
antifraud_*	A..100	O	Дополнительные параметры для системы антифрод. Можно передать любые параметры, начинающиеся с antifraud_
user_entered_*	A..100	O	Можно указать дополнительные поля для оплаты, начинающиеся с user_entered_. Затем все эти поля

Название	Тип	R/O	Описание
			возвращаются в GetAdvancedStatus в поле user_entered_params
payer_ip	A	R	IP-адрес браузера покупателя
browserAccept	A	R	Заголовок Accept из браузера покупателя
browserLanguage	A	R	Язык браузера устройства покупателя. en — английский; ru — русский
browserUserAgent	A	R	Содержание заголовка User-Agent в браузере покупателя
browserJavaEnabled	B	R	Признак возможности выполнения JavaScript в браузере устройства покупателя
browserScreenHeight	A	R	Общая высота экрана устройства покупателя в пикселях
browserScreenWidth	A	R	Общая ширина экрана устройства покупателя в пикселях
browserColorDepth	A	R	Глубина цветопередачи в битах
browserTZ	A	R	Часовой пояс — разница (в минутах) во времени между временем по UTC и местным временем браузера на устройстве покупателя
threeDsNotificationUrl	A	R	Url перенаправления после выполнения ChallengeComplete

Параметры ответа:

Формат ответа совпадает форматом ответа [PayMerchant](#)

GetStatusSaveCard

Запрос статуса сохранённой карты.

Формат запроса: <https://{{host}}.payler.com/{{api}}/GetStatusSaveCard>

Доступно для: gapi, mapi, cgapi, cmapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
session_id	A..100	O	Идентификатор сессии, по которой требуется получить информацию. Если не указан, должен быть указан параметр card_id
card_id	A..100	O	Идентификатор карты, по которой требуется получить информацию. Если не указан, должен быть указан параметр session_id

Параметры ответа:

Название	Тип	R/O	Описание
card_id	A..100	R	Идентификатор карты
card_status	A	R	Статус карты. <ul style="list-style-type: none"> • Saved — карта сохранена и готова к использованию. • Invalid — карта недействительна (например, истёк срок её действия)
card_number	A..19	R	Маскированный номер карты. Оставлены первые 6 и последние 4 цифры, остальные заменены на 'x'
card_holder	A..100	R	Имя владельца карты
expired_year	N2	R	Год истечения срока действия карты
expired_month	N2	R	Месяц истечения срока действия карты
recurrent_template_id	A..100	O	Идентификатор шаблона рекуррентных платежей
customer_id	A..100	R	Идентификатор пользователя

Пример ответа на успешный запрос:

```
{
    "card_status": "Saved",
    "card_id": "ZRzroTofO42UkBv5vsMmckgU3Dowz7J4UhMZ",
    "card_number": "400000xxxxxx0093",
    "card_holder": "NONAME",
    "expired_year": 25,
    "expired_month": 12,
    "customer_id": "BCaFGEp8gQhT1XXFs1HEf5LRFstC4efydqxv"
}
```

GetCardList

Запрос на получение списка сохраненных карт.

Формат запроса: <https://{{host}}.payler.com/{{api}}/GetCardList>

Доступно для: gapi, mapi, cgapi, cmapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
customer_id	A..100	R	Идентификатор пользователя. Будут возвращены карты, принадлежащие этому пользователю

Параметры ответа:

Название	Тип	R/O	Описание
cards	Ar	R	Список карт
cards/card_id	A..100	R	Идентификатор карты
cards/customer_id	A..100	R	Идентификатор пользователя
cards/card_status	A	R	Статус карты. <ul style="list-style-type: none"> Saved — карта сохранена и готова к использованию. Invalid — карта недействительна (например, истёк срок её действия)
cards/card_number	A..19	R	Маскированный номер карты. Оставлены первые 6 и последние 4 цифры, остальные заменены на 'x'
cards/card_holder	A..100	R	Имя владельца карты
cards/expired_year	N2	R	Год истечения срока действия карты
cards/expired_month	N2	R	Месяц истечения срока действия карты
cards/recurrent_template_id	A..100	O	Идентификатор шаблона рекуррентных платежей. Присутствует если по карте создан шаблон рекуррентных платежей

Пример ответа на успешный запрос:

```
{
    "cards": [
        {
            "card_status": "Saved",
            "card_id": "Zk8ou8tcfiitc31DwJPOIIowC8oFZO1IDknL",
            "customer_id": "jsdlf8sDD9Lh019sVYotkJ2gHSPs5E4Qph14",
            "card_number": "452427xxxxxx1232",
            "card_holder": "IVAN PETROV",
            "expired_year": 17,
            "expired_month": 7
        },
        {
            "card_status": "Saved",
            "card_id": "OvzLAlY3twt5GYnm2ddGiixhaltNSHJcXk2k",
            "customer_id": "sdkKhnf23ds987K8bsdj791dn23fsldmMJ",
            "card_number": "415432xxxxxx8564",
            "card_holder": "PETR IVANOV",
            "expired_year": 16,
            "expired_month": 9
        }
    ]
}
```

RemoveCard

Запрос удаления сохранённой карты.

URL запроса: <https://{{host}}.payler.com/{{api}}/RemoveCard>

Доступно для: gapi, mapi, csgapi, cmapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
card_id	A..100	R	Идентификатор карты по которой требуется получить информацию

Параметры ответа:

Название	Тип	R/O	Описание
changed	B	R	Показывает, привёл ли запрос к изменениям в системе. 1 — карта была удалена в результате этого запроса. 0 — карта была удалена ранее

Пример ответа на успешный запрос:

```
{
    "changed": true
}
```

SaveCard

Запрос сохранения карточных данных, переданных через API, без перенаправления на страницу платёжного шлюза Payler.

Формат запроса: <https://{{host}}.payler.com/{{api}}/v1/SaveCard>

Доступно для: мари, стари

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
customer_id	A..100	R	Идентификатор плательщика. Возвращается в ответ на запрос CustomerRegister
card_number	A..19	R	Номер карты. Десятичные цифры без разделителей [0–9]
card_holder	A..100	O	Имя владельца карты. Латинские символы, десятичные цифры или символ пробела [a-zA-Z0-9]
expired_year	N2	R	Год окончания срока действия карты
expired_month	N2	R	Месяц окончания срока действия карты. Целое двухзначное число в формате с лидирующим нулем
lang	A2	O	Предпочитаемый язык платёжной формы и ответов сервера. en — английский; ru — русский. По умолчанию — ru

Параметры ответа:

Название	Тип	R/O	Описание
card_id	A..100	R	Идентификатор карты
card_status	A	R	Статус карты. <ul style="list-style-type: none"> • Saved — карта сохранена и готова к использованию • Invalid — карта недействительна (например, истёк срок её действия)
card_number	A..19	R	Маскированный номер карты. Оставлены первые 6 и последние 4 цифры, остальные заменены на 'x'
card_holder	A..100	R	Имя владельца карты
expired_year	N2	R	Год истечения срока действия карты
expired_month	N2	R	Месяц истечения срока действия карты

Пример ответа на успешный запрос:

```
{
    "card_id": "z5RkjN4OjQ6I1UwTlpHJKBKj5Kf99D4T8r1c",
    "card_status": "Saved",
    "customer_id": "wsdj24DDgs34fsfDfs24Dr45L98k1km9kknc",
    "card_number": "400000xxxxxx0002",
    "card_holder": "NONAME",
    "expired_year": 18,
    "expired_month": 7
}
```

CustomerRegister

Запрос регистрации нового пользователя.

Формат запроса: <https://{{host}}.payler.com/{{api}}/CustomerRegister>

Доступно для: gapi, mapi, cgapi, cmapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
customer_name	A..100	O	Имя пользователя
customer_phone	A..100	O	Номер телефона пользователя
customer_email	A..100	O	Адрес электронной почты пользователя
customer_fullName	A..100	O	Ф. И. О. пользователя
customer_address	A..100	O	Адрес пользователя
customer_documentType	A..60	O	Тип документа пользователя
customer_documentSerial	A..60	O	Серия документа пользователя
customer_documentNumber	A..60	O	Номер документа пользователя

Параметры ответа:

Название	Тип	R/O	Описание
customer_id	A..100	R	Идентификатор пользователя

Пример ответа на успешный запрос:

```
{
    "customer_id": "Wkl7nBsMhwQICc2I9wXncMeSwnIaucQ5BHy0"
}
```

CustomerUpdate

Запрос обновления данных о пользователе.

Формат запроса: <https://{{host}}.payler.com/{{api}}/CustomerUpdate>

Доступно для: gapi, mapi, cgapi, cmapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
customer_id	A..100	R	Идентификатор пользователя
customer_name	A..100	O	Имя пользователя
customer_phone	A..100	O	Номер телефона пользователя
customer_email	A..100	O	Адрес электронной почты пользователя
customer_fullName	A..100	O	Ф. И. О. пользователя
customer_address	A..100	O	Адрес пользователя
customer_documentType	A..60	O	Тип документа пользователя
customer_documentSerial	A..60	O	Серия документа пользователя
customer_documentNumber	A..60	O	Номер документа пользователя

Параметры ответа:

Название	Тип	R/O	Описание
customer_id	A..100	R	Идентификатор пользователя

Пример ответа на успешный запрос:

```
{
    "customer_id": "Wk17nBsMhwQICc2I9wXncMeSwnIaucQ5BHyo"
}
```

CustomerDelete

Запрос удаления пользователя.

Формат запроса: <https://{{host}}.payler.com/{{api}}/CustomerDelete>

Доступно для: gapi, mapi, cgapi, cmapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
customer_id	A..100	R	Идентификатор пользователя

Параметры ответа:

Название	Тип	R/O	Описание
changed	B	R	Показывает, привёл ли запрос к изменениям в системе. 1 — плательщик был удалён в результате этого запроса, 0 — плательщик был удалён ранее

Пример ответа на успешный запрос:

```
{
    "changed": true
}
```

CustomerGetStatus

Проверка регистрации пользователя в системе.

Формат запроса: <https://{{host}}.payler.com/{{api}}/CustomerGetStatus>

Доступно для: gapi, mapi, cgapi, cmapi

Параметры запроса:

Название	Тип	R/O	Описание
key	A..100	R	Идентификатор мерчанта. Выдается с параметрами доступа
customer_id	A..100	R	Идентификатор пользователя

Параметры ответа:

Название	Тип	R/O	Описание
status	A..10	R	Показывает, активен ли пользователь. Active — пользователь зарегистрирован в системе. Если пользователь не зарегистрирован, то в ответ на запрос вернется ошибка

Пример ответа на успешный запрос:

```
{
    "status": "Active"
}
```

Ответы в случае неудачных запросов

В случае неудачного завершения запроса ответ будет содержать объект `error` с описанием ошибки.

Параметры описания ошибки:

Название	Тип	R/O	Описание
code	N	R	Внутренний код ошибки. Необходим при обращении к службе технической помощи
message	A	R	Описание возникшей ошибки

Пример ответа в случае неудачного запроса:

```
{
    "error": {
        "code": 14,
        "message": "Unable to perform the operation within the non two-step payment"
    }
}
```

Коды ошибок сервера

Приведенный ниже список является исчерпывающим на момент составления. Тем не менее, сервер постоянно обновляется и при написании пользовательского кода следует учитывать, что может прийти отсутствующая в списке ошибка.

Код	Имя	Описание
0	NONE	Ошибки нет
1	INVALID_AMOUNT	Неверно указана сумма транзакции
2	BALANCE_EXCEEDED	Превышен баланс карты
3	DUPLICATE_ORDER_ID	Попытка создать сессию оплаты с идентификатором заказа, для которого сессия уже была зарегистрирована ранее
4	ISSUER_DECLINED_OPERATION	Банк-эмитент карты отказал в операции. Для получения детальной информации требуется связаться с его службой поддержки
5	LIMIT_EXCEEDED	Достигнуты ограничительные условия по карте
6	AF_DECLINED	Операция отклонена системой безопасности. Требуется обратиться в службу поддержки или использовать другую карту
7	INVALID_ORDER_STATE	Операция не может быть выполнена, так как платеж находится в недопустимом состоянии
9	ORDER_NOT_FOUND	Сессия платежа с указанным идентификатором не найдена.
10	PROCESSING_ERROR	Ошибка общего характера, возникшая при взаимодействии с процессинговым центром банка. Требуется связаться со службой поддержки.
11	PARTIAL_RETRIEVE_NOT_ALLOWED	Частичная разблокировка не поддерживается (сумма авторизации не может быть изменена)
13	GATE_DECLINED	Отказ шлюза в выполнении транзакции. Требуется связаться со службой поддержки
14	INVALID_CARD_INFO	Указаны некорректные данные карты

Код	Имя	Описание
15	INVALID_CARDNUMBER	Указан некорректный номер карты
16	INVALID_CARDHOLDER	Указано некорректное имя держателя карты
18	API_NOT_ALLOWED	Данный метод Payler API не разрешен к использованию с IP-адреса, с которого выполнен запрос
19	INVALID_PASSWORD	Доступ запрещен, так как пароль некорректен
20	INVALID_PARAMS	Один из параметров запроса имеет некорректное значение
21	SESSION_TIMEOUT	Время, отведенное на осуществление платежа, истекло
22	MERCHANT_NOT_FOUND	Описание мерчанта не найдено
24	SESSION_NOT_FOUND	Сессия платежа не найдена
25	CARD_EXPIRED	Срок действия карты истек
26	RECURRENT_TEMPLATE_NOT_FOUND	Шаблон рекуррентных платежей с указанным идентификатором не найден
27	RECURRENT_TEMPLATE_NOT_ACTIVE	Шаблон рекуррентных платежей неактивен
28	NO_TRANSACTION_BY_TEMPLATE	Ранее не было выполнено транзакций по шаблону
100	RECURRENT_PAYMENTS_NOT_SUPPORTED	Рекуррентные платежи не поддерживаются. Для включения данной функциональности требуется обратиться в службу поддержки
101	EXPIRED_RECURRENT_TEMPLATE	Срок действия шаблона рекуррентных платежей истек
102	RECURRENT_TEMPLATE_ANOTHER_TERMINAL	Шаблон рекуррентных платежей зарегистрирован на другой терминал
103	FAILED_UPDATE_ACTIVE_STATUS	Не удалось обновить статус активности шаблона рекуррентных платежей. Требуется связаться со службой поддержки
104	TEMPLATE_ACTIVATIONQUIRES_BANK_CONF	Активация шаблона рекуррентных платежей требует подтверждения со стороны банка
105	REFUND_OF_RECURRENT_NOT_SUPPORTED	Возвраты рекуррентных платежей не поддерживаются банком

Код	Имя	Описание
106	TOO_FREQUENT_RECURRENT_PAYMENTS	Слишком частые рекуррентные платежи
200	PARTIAL_REFUND_NOT_ALLOWED	Частичный возврат не поддерживается (невозможно вернуть пользователю часть ранее списанной суммы)
201	MULTIPLE_REFUND_NOT_SUPPORTED	Последующие возвраты не поддерживаются в автоматическом режиме. Требуется обратиться в службу поддержки для их выполнения
202	PARTIAL_CHARGE_NOT_ALLOWED	Частичное списание не поддерживается
300	EXPIRED_RETRIEVE_PERIOD	Истек период изменения суммы заблокированных средств
400	INVALID_EXPIRY_MONTH	Указано некорректное значение месяца истечения срока действия карты
401	INVALID_EXPIRY_YEAR	Указано некорректное значение года истечения срока действия карты
402	INVALID_SECURE_CODE	Указано некорректное значение кода подлинности карты
403	INVALID_EMAIL	Указано некорректное значение адреса электронной почты
500	CARD_INACTIVE	Карта неактивна. Требуется оплатить другой картой или связаться со службой поддержки банка-эмитента карты
501	OPERATION_NOT_SUPPORTED	Операция не поддерживается картой. Требуется связаться со службой поддержки банка-эмитента карты
502	DECLINED_BY_CARDHOLDER	Операция отменена по желанию держателя карты
503	PIN_ERROR	Ошибка обработки PIN-кода карты. Требуется оплатить другой картой или связаться со службой поддержки банка-эмитента карты
504	RESTRICTED_CARD	Ограниченнная карта. Требуется оплатить другой картой или связаться со службой поддержки банка-эмитента карты
505	INVALID_CARD_STATUS	Неверный статус карты
600	DUPLICATED_OPERATION	Повторная операция. Требуется связаться со службой поддержки
601	IN_PROGRESS_ERROR	Запрос уже обрабатывается

Код	Имя	Описание
602	PAID_EARLIER	Заказ был оплачен ранее
603	DEAL_NOT_FOUND	Не найдено зарегистрированных транзакций по указанному идентификатору заказа (пользователь не предпринимал попыток оплаты)
604	INCORRECT_TRANSACTION_TYPE	Транзакция имеет неподдерживаемый для текущей операции тип
605	TRANSACTION_NOT_TWO_STEP	Невозможно выполнить операцию в рамках не двухстадийного платежа
606	ATTEMPT_NOT_FOUND	Не найдено описание попытки платежа с указанным идентификатором. Требуется связаться со службой поддержки
607	ATTEMPTS_NUMBER_EXCEEDED	Превышено максимальное число попыток осуществления платежа
608	THERE_IS_NEWER_ATTEMPT	Начата новая попытка оплаты, продолжение старой невозможно
609	EMAIL_ATTEMPTS_NUMBER_EXCEEDED	Превышено максимальное число попыток отправить письмо
610	CARD_NOT_FOUND	Карта не найдена
611	CARD_ALREADY_SAVED	Данная карта уже сохранена
612	CUSTOMER_NOT_FOUND	Пользователь не найден
700	TEMPLATE_NOT_FOUND	Невозможно создать указанный шаблон ответа
701	RETURN_URL_NOT_SET	Не задан URL перенаправления на сайт мерчанта после осуществления платежа
702	TERMINAL_NOT_FOUND	Терминал мерчанта с указанным платежным идентификатором не найден
703	CURRENCY_NOT_SUPPORTED	Данная валюта не поддерживается
704	RECEIPT_SERVICE_NOT_ENABLED	Сервис касс не подключен или не активен. Пожалуйста, обратитесь в службу поддержки для включения
800	THREE_DS_FAIL	3D-Secure аутентификация на стороне банка-эмитента карты отменена, или ее выполнение невозможно из-за возникшей ошибки
801	NO_RESULT_OF_3DS	Не найдено описание результата

Код	Имя	Описание
		попытки осуществления платежа, требующего 3D-Secure подтверждения. Требуется связаться со службой поддержки
802	PREPROCESS_3DS_INFO_NOT_FOUND	Информация о предварительной обработке 3D-Secure запроса не найдена. Требуется связаться со службой поддержки
803	NOT_INVOLVED_IN_3DS	Используемая карта не поддерживает 3D-Secure аутентификацию. Требуется оплатить картой с поддержкой технологии 3D-Secure
900	OPERATION_NOT_ALLOWED_TO_MERCHANT	Операция не разрешена мерчанту. Требуется связаться со службой поддержки
901	COMPLETED_PARTIALLY	Операция выполнена не полностью. Требуется связаться со службой поддержки
902	RECONCILE_ERROR	Сверка итогов завершилась с ошибкой. Требуется связаться со службой поддержки
903	DECLINED	Операция отклонена. Требуется связаться со службой поддержки
904	TEMPORARY_MALFUNCTION	Временная проблема. Попробуйте выполнить операцию позднее
905	UNSUPPORTED_CARD_TYPE	Операции по данному типу карт не поддерживаются
1000	EMONEY_REFUND_NOT_SUPPORTED	Для платежей через электронный кошелек возврат не поддерживается
1100	CHECKOUT_METHOD_NOT_ALLOWED	Метод оплаты checkout не разрешен для данного мерчанта
1101	OPERATION_NOT_CONFIRMED	Мерчант не подтвердил операцию
1102	AMOUNT_LIMIT_EXCEEDED	Превышена максимальная сумма операции