



Протокол взаимодействия мастер мерчанта и системы PSP Pay-logic.

Программное обеспечение «PSP Pay-logic».

Руководство разработчика

АННОТАЦИЯ

| Содержит описание принципов работы, настройки и интеграции в системы партнеров

Версия руководства: 1.3

Руководство актуально для кабинета мерчанта версий 1.14

2008–2026 ООО «Софт-Лоджик», г. Барнаул, Россия

Данный документ входит в комплект поставки программных продуктов.

Права использования данного документа предусмотрены соответствующим лицензионным договором.

ООО «Софт-Лоджик»

656006, г. Барнаул, Малахова ул., дом 146в

Тел: (3852) 72-27-27

© *Soft-logic*

Web: <https://soft-logic.ru/>

Mail: info@soft-logic.ru

ОГЛАВЛЕНИЕ

ИСТОРИЯ ИЗМЕНЕНИЙ	5
ВЕРСИЯ ДОКУМЕНТА 1.0	5
ВЕРСИЯ ДОКУМЕНТА 1.1	5
ВЕРСИЯ ДОКУМЕНТА 1.2	5
ВЕРСИЯ ДОКУМЕНТА 1.3	6
1 ТЕРМИНЫ И СОКРАЩЕНИЯ	7
2 ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ	8
3 ВВЕДЕНИЕ	9
4 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ	10
4.1 СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ	10
4.1.1 ВЗАИМНЫЙ TLS (MUTUAL TLS).....	10
4.2 АВТОРИЗАЦИЯ	11
5 ИНТЕГРАЦИЯ В СИСТЕМУ ПАРТНЕРА	12
5.1 ОБЩАЯ ИНФОРМАЦИЯ	12
5.2 ЗАПРОС ИНФОРМАЦИИ ИЗ QR-КОДА	13
5.3 ЗАПРОС СТАТУСА ОПЕРАЦИИ	16
5.4 ОТМЕНА ПЛАТЕЖА	18
5.4.1 ЗАПРОС НА ОТМЕНУ ПЛАТЕЖА И ВОЗВРАТ СРЕДСТВ.....	19
5.4.2 ЗАПРОС СТАТУСА ЗАЯВКИ НА ОТМЕНУ ПЛАТЕЖА.....	21
5.5 УВЕДОМЛЕНИЕ ПАРТНЕРА О СТАТУСЕ ОПЕРАЦИИ	23
6 ПРИЛОЖЕНИЕ	28
А. ПРИМЕРЫ ГЕНЕРАЦИИ И ПРОВЕРКИ ЭП НА JAVA	28
В. ПРИМЕР ЗАПРОСА И ОТВЕТА ДЛЯ МЕТОДА ЗАПРОСА ИНФОРМАЦИИ ИЗ QR-КОДА	30
С. ПРИМЕР ЗАПРОСА И ОТВЕТА ДЛЯ МЕТОДА ЗАПРОСА СТАТУСА ЗАКАЗА	30

D. ПРИМЕР ЗАПРОСА И ОТВЕТА ДЛЯ ОПЕРАЦИИ ОТМЕНЫ ПЛАТЕЖА И ВОЗВРАТА СРЕДСТВ.....	32
E. ПРИМЕР ЗАПРОСА И ОТВЕТА ДЛЯ МЕТОДА ЗАПРОСА ЗАЯВКИ НА ОТМЕНУ ПЛАТЕЖА.....	32
F. ПРИМЕР WEBНООК-ОПОВЕЩЕНИЯ.....	33
G. СТАТУСЫ ОПЕРАЦИЙ.....	34

ИСТОРИЯ ИЗМЕНЕНИЙ**ВЕРСИЯ ДОКУМЕНТА 1.0**

Дата публикации: 03.03.2026 г.

Изменение	Раздел
Документ создан	-

ВЕРСИЯ ДОКУМЕНТА 1.1

Дата публикации: 09.04.2026 г.

Изменение	Раздел
Изменения в документе:	
Дополнено описание технических требований	4
Дополнено описание авторизации в уведомлениях партнеров о статусе оплаты	5.4

ВЕРСИЯ ДОКУМЕНТА 1.2

Дата публикации: 14.04.2026 г.

Изменение	Раздел
Общие улучшения:	
Актуализированы примеры генерации и проверки электронных	Приложение А

Изменение	Раздел
подписей	

ВЕРСИЯ ДОКУМЕНТА 1.3

Дата публикации: 23.04.2026 г.

Изменение	Раздел
Общие улучшения:	
Добавлено описание функционала возврата средств	5.4

1 ТЕРМИНЫ И СОКРАЩЕНИЯ

Мерчант — юридическое лицо, занимающееся продажей товаров или предоставлением услуг и принимающее оплату через электронные платежные системы.

Мастер мерчант (Master Merchant) — это юридическое лицо-посредник, который агрегирует платежный трафик группы организаций-продавцов, называемых Суб-мерчанты (Sub-Merchants), и берет на себя все технические сложности приема платежей.

MID — идентификатор продавца.

QR-код — двухмерный матричный штрихкод, который выступает в роли графического носителя платежной информации. Содержит зашифрованные реквизиты получателя средств (например, идентификатор точки, номер счета, сумму транзакции и т.д.) и служит для инициирования безналичного перевода денежных средств.

Мульти QR — технология приема оплаты по QR-кодам в пользу зарубежных мерчантов через мобильное приложение СберБанка.

ТСТ — торгово-сервисная точка.

ЭП — электронная подпись.

2 ПРЕДПОЛАГАЕМАЯ АУДИТОРИЯ

Данное руководство предназначено для использования в качестве спецификации для настройки и интеграции мастер мерчанта и PSP Pay-logic.

Документ адресован IT-специалистам (программистам и администраторам), обслуживающим информационную систему поставщика услуг и знакомым с технологиями разработки ПО, языками программирования, основами работы с XML и сетевым взаимодействием по HTTP/HTTPS, владеющим основами криптографии.

Необходимо знание языков программирования.

3 ВВЕДЕНИЕ

Документ предназначен для систем мастер мерчантов и их разработчиков:

1. Для проведения доработки (разработки) программного обеспечения для систем мастер мерчантов, разработки технических заданий на доработку (разработку) соответствующих систем, обеспечивающих возможность подключения к системам мастер мерчантов.
2. Для оценки качества проведенных доработок и соответствия доработанной системы общим техническим требованиям во время комплексных испытаний. Реализация требований данного документа позволит произвести подключение к системе PSP и обеспечить проведение операций с денежными средствами.

На настоящий момент протокол, описываемый данным документом, поддерживает только способ оплаты Мульти QR.

В рамках данного документа понятие мерчант используется в контексте подключения мастер мерчанта к системе PSP Paylogic.

4 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

4.1 СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ

Сетевое взаимодействие системы мастер мерчанта и PSP должно происходить в формате JSON по протоколу HTTPS. Для взаимодействия с системой мастер мерчанта PSP предоставляет адрес, на который будут отправляться запросы. HTTP-метод каждого запроса определяется описанием соответствующего API-метода, возможны GET и POST.

Sandbox Base URL: `https://{host1}:{port1}`

Production Base URL: `https://{host2}:{port2}`

IP whitelist: не требуется. Возможно использование по согласованию сторон.

Поддерживаемые версии TLS: 1.2, 1.3

Значения параметров сетевого взаимодействия (host1, port1, host2, port2) необходимо дополнительно уточнить у курирующего менеджера продукта.

4.1.1 ВЗАИМНЫЙ TLS (MUTUAL TLS)

По согласованию сторон для сетевого взаимодействия возможно использование mTLS. В этом случае используются отдельные сертификаты для sandbox и production.

Запрос на подпись сертификата и обмен сертификатами осуществляется через курирующего менеджера проекта по согласованию.

4.2 АВТОРИЗАЦИЯ

В системе для авторизации запросов используется проверка электронной подписи. Для этого система ожидает передачу специальных HTTP-заголовков:

- **PSP-Point** — идентификатор API мастер мерчанта, присвоенный в системе Paylogic;
- **PSP-Sign** — подпись запроса.

Электронная подпись формируется от строки `<REQUEST METHOD>+<REQUEST FULL URI>+<REQUEST BODY>` по алгоритму SHA256withRSA в формате Base64 (UTF8). Для GET и иных запросов без тела **BODY** принимается равным пустой строке.

Ключи для формирования подписи генерируются через кабинет пользователя PSP при создании API с типом «Host to Host sign». При этом ключи не хранятся на сервере PSP и доступны для сохранения только на компьютере пользователя.

Для sandbox и prodaction используются разные профили ключей. Порядок ротации, замены и отзыва ключей определяется по согласованию сторон.

Пример подписываемой строки:

```
POST/psp/external/api/payment/status?id=external-id-123321
```

Получившаяся подпись будет иметь вид:

```
vhF2ZzOYR1tdnASpbd1tCXCrPh3IdLo2vhB133fkvEX7vNSkVvkT2G+sqf2jMAUTQ7c/T2s/  
ppuz/7aLXZWAQbAPYIatgRXIwXSJW8y8SO99h4jQrenVly1abPLSgC3SK0scmhBwkMGA/iI/  
i/5S+q+tIfDcnbCRltI1We9Gi3A=
```

Пример генерации и проверки подписи приведен в приложении [А. Примеры генерации и проверки ЭП на java.](#)

5 ИНТЕГРАЦИЯ В СИСТЕМУ ПАРТНЕРА

5.1 ОБЩАЯ ИНФОРМАЦИЯ

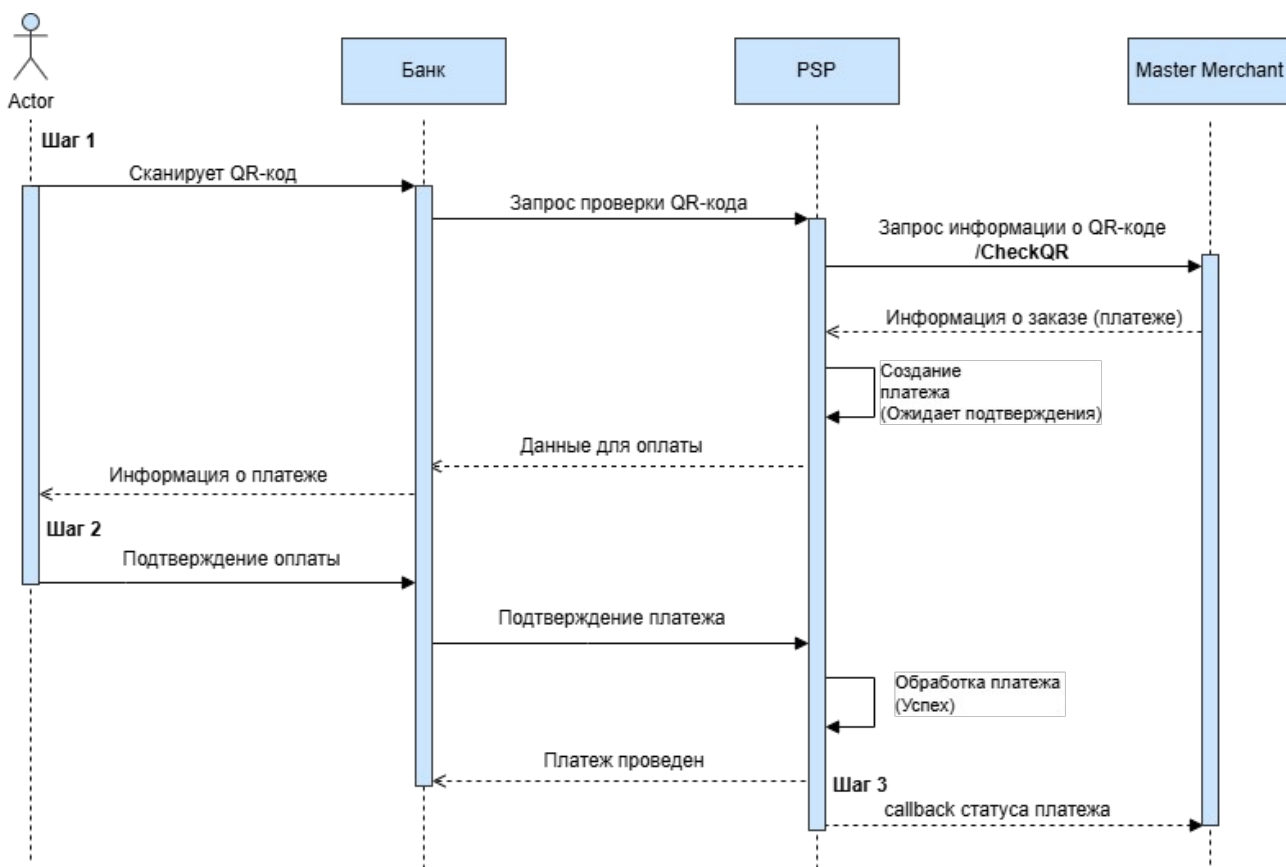


Рисунок 5.1.1 — Диаграмма взаимодействия участников процесса оплаты по QR-коду

Шаг 1: Клиент сканирует QR-код через приложение банка. Банк отправляет данные QR в PSP для проверки QR-кода и получения информации о заказе. PSP, в свою очередь, делает запрос к мастер мерчанту для получения данных заказа, далее PSP создает платеж в статусе «Ожидает подтверждения» и возвращает ответ банку для проведения платежа на его стороне.

Шаг 2: Клиент подтверждает оплату в приложении банка. Банк отправляет запрос в PSP на подтверждение заказа. PSP помечает заказ как оплаченный.

Шаг 3: PSP отправляет callback-нотификацию статуса платежа мастер мерчанту.

Дополнительные возможности:

- При необходимости мастер мерчантом может быть дополнительно запрошен статус заказа вызовом метода `/psp/external/api/payment/status`. Метод вернет в ответ текущий статус заказа (описание метода и его параметров приведено в разделе 5.3). Рекомендуется запросить статус если автоматически не был получен ответ в течении 5 минут.
- В случае, если PSP не получила от банка подтверждение оплаты по QR-коду в течение заданного промежутка времени (по умолчанию 10 мин.), то платеж на стороне PSP будет отменен и мастер мерчанту будет отправлена нотификация об отмене платежа. Точные настройки тайм-аута ожидания подтверждения оплаты от банка необходимо уточнить у менеджера проекта.

5.2 ЗАПРОС ИНФОРМАЦИИ ИЗ QR-КОДА

Для получения информации от мастер мерчанта о платежном запросе, зашифрованном в QR-коде, используется POST метод `https://{host}/CheckQR`.

Структура запроса:

```
{
  "OperationTime": "YYYY-MM-DDThh24:mm:ss.ff3TZH:TZM"
  "QrData": "string"
}
```

Таблица 5.2.1 — Параметры запроса информации из QR-кода

Параметр	Описание	Обяз.
OperationTime	Дата и время формирования запроса по ISO 8601. Формат: ГГГГ-ММ-ДДТчч:мм:сс.ссс±чч:мм	Да
QrData	Строка, зашифрованная в QR-коде. Передается без изменений, без парсинга	Да

Структура ответа:

```
{
  "status": "string",
  "errorCode": "string",
  "errorMessage": "string",
  "data": {
    "id": "string",
    "merchantName": "string",
    "merchantID": "string",
    "country": "string",
    "city": "string",
    "merchantAddress": "string",
    "mcc": "6543",
    "terminalId": "string",
    "amount": 100000,
    "currency": "string",
  }
}
```

Таблица 5.2.2 — Параметры ответа

Параметр	Описание	Обяз.
status	Статус обработки запроса: <ul style="list-style-type: none">• SUCCESS — запрос обработан успешно;• ERROR — ошибка обработки запроса, необходимо анализировать поле	Да

	errorCode	
errorCode	Детализирующий код ошибки обработки запроса, возможные значения приведены в таблице <u>G.2</u>	Да
errorMessage	Текстовое описание ошибки	Нет
data[]	Объект данных, полученных из QR-кода	Да
data[].id	Уникальный идентификатор заказа в системе мастер мерчанта	Да
data[].merchantName	Наименование ТСТ	Да
data[].merchantID	Уникальный идентификатор ТСТ	Да
data[].country	Трехзначный код страны согласно ISO 3166-1 numeric	Да
data[].city	Наименование населенного пункта	Да
data[].merchantAddress	Адрес ТСТ	Да
data[].mcc	Код категории продавца (Merchant Category Code)	Да
data[].terminalId	Идентификатор терминала, сгенерировавшего QR-код	Да

<code>data[].amount</code>	Сумма заказа в минорной единице валюты (копейки, центы и т.д.)	Да
<code>data[].currency</code>	Код валюты заказа по стандарту ISO 4217	Да

Примеры запроса и ответов приведены в приложении В. Пример запроса и ответа для метода запроса информации из QR-кода.

5.3 ЗАПРОС СТАТУСА ОПЕРАЦИИ

GET метод `/psp/external/api/payment/status` предназначен для запроса статуса заказа в системе.

Запрос имеет один параметр: `id`, уникальный идентификатор заказа.

Параметр является обязательным, тип данных — `string`.

Структура ответа:

```
{
  "attributes": [
    {
      "code": "string",
      "flags": 0,
      "title": "string",
      "value": "string",
      "valueTitle": "string"
    }
  ],
  "bankPayment": {
    "attributes": [
      {
        "code": "string",
        "flags": 0,
```

```
        "title": "string",
        "value": "string",
        "valueTitle": "string"
    }
],
"error": 0,
"errorMessage": "string",
"status": "ERROR",
"transaction": "string"
},
"date": "string",
"error": 0,
"errorDetail": {
    "errorFields": [
        "string"
    ],
    "message": "string",
    "type": "COMMISSION_ERROR"
},
"id": "string",
"menu": {
    "category": "string",
    "service": 0,
    "serviceName": "string"
},
"message": "string",
"status": "ALREADY_EXISTS",
"transaction": 0
}
```

Таблица 5.3.1 — Атрибуты ответа на запрос статуса заказа

Параметр	Описание	Обяз.
id	Уникальный идентификатор заказа на стороне мастер мерчанта	Да
status	Статус заказа. Возможные статусы приведены в таблице G.1	Да

bankPayment.status	Статус банковской операции. Возможные статусы приведены в таблице G.1	Нет
bankPayment.transaction	Уникальный идентификатор транзакции банка	Нет
date	Дата создания операции, в формате уууу-ММ dd'T'HH:mm:ss	Нет
attributes	Объект данных атрибутов основной операции. Описание объекта приведено в таблице 5.5.3	Нет
menu	Объект данных сервиса, по которому осуществлялась оплата. Описание объекта приведено в таблице 5.5.4	Нет
transaction	Уникальный идентификатор транзакции на стороне PSP	Нет

Примеры запроса и ответа приведены в приложении [С. Пример запроса и ответа для метода запроса статуса заказа](#).

5.4 ОТМЕНА ПЛАТЕЖА

В системе предусмотрена процедура отмены совершенного платежа и возврата средств. Функционал возврата включает в себя 2 метода:

1. **/psp/external/api/payment/cancel/process** — создание заявки на отмену платежа и возврат денежных средств;

2. **/psp/external/api/v2/payment/cancel/status** — получение статуса созданной заявки на отмену платежа.

5.4.1 ЗАПРОС НА ОТМЕНУ ПЛАТЕЖА И ВОЗВРАТ СРЕДСТВ

Для создания запроса на отмену или возврат средств по платежу мастер мерчант вызывает метод **POST /psp/external/api/payment/cancel/process**.

Структура запроса:

```
{
  "id": "string",
  "comment": "string",
  "sum": 0,
  "currency": "string"
}
```

Таблица 5.4.1.1 — Атрибуты запроса на отмену и возврат средств по платежу

Параметр	Описание	Обяз.
id	Уникальный идентификатор заказа на стороне мастер мерчанта	Да
comment	Комментарий	Нет
sum	Сумма возврата, в минимальных единицах валюты	Нет
currency	Валюта суммы возврата	Нет

Структура ответа:

```
{
  "id": "string",
```

```
"transaction": 0,  
"sum": 0,  
"currency": "string",  
"comment": "string",  
"date": "2026-04-23T02:21:15.328Z",  
"status": "SUCCESS"  
}
```

Таблица 5.4.1.2 — Атрибуты ответа на запрос отмены платежа и возврат средств

Параметр	Описание	Обяз.
id	Уникальный идентификатор заказа на стороне мастер мерчанта	Да
transaction	Уникальный идентификатор транзакции на стороне PSP	Нет
sum	Сумма возврата, в минимальных единицах валюты	Нет
currency	Валюта суммы возврата	Нет
comment	Комментарий	Нет
date	Дата и время создания или последнего изменения состояния операции возврата, в формате уууу-ММ dd'T'HH:mm:ss	Нет
status	Статус обработки операции возврата	Нет

Пример запроса и ответа приведены в приложении Д. Пример запроса и ответа для операции отмены платежа и возврата средств.

5.4.2 ЗАПРОС СТАТУСА ЗАЯВКИ НА ОТМЕНУ ПЛАТЕЖА

Для получения статуса заявки на отмену и возврат платежа мастер мерчант вызывает метод **GET/psp/external/api/v2/payment/cancel/status**.

В качестве параметров запроса передается **id**, уникальный идентификатор заказа. Параметр является обязательным, тип данных — string.

Структура ответа:

```
{
  "error": 0,
  "errorMessage": "string",
  "id": "string",
  "transaction": 0,
  "status": "SUCCESS",
  "sum": 0,
  "currency": "string",
  "rejectRequests": [
    {
      "requestId": 0,
      "sum": 0,
      "currency": "string",
      "comment": "string",
      "date": "2026-04-23T03:41:23.324Z",
      "status": "SUCCESS"
    }
  ]
}
```

Таблица 5.4.2.1 — Атрибуты ответа на запрос статуса заявки на отмену платежа

Параметр	Описание	Обяз.
error	Код ошибки	Да
errorMessage	Текстовое описание ошибки	Нет

<code>id</code>	Уникальный идентификатор заказа на стороне мастер мерчанта	Да
<code>transaction</code>	Уникальный идентификатор операции на стороне PSP	Да
<code>status</code>	Статус обработки запроса	Да
<code>sum</code>	Сумма заказа, с минимальных единиц валюты	Да
<code>currency</code>	Валюта заказа	Нет
<code>rejectRequests []</code>	Массив данных заявок на возврат	Да
<code>rejectRequests [].requestId</code>	Уникальный идентификатор операции возврата	Да
<code>rejectRequests [].sum</code>	Сумма возврата, в минимальных единицах валюты	Да
<code>rejectRequests [].currency</code>	Валюта возврата	Нет
<code>rejectRequests [].comment</code>	Комментарий	Нет
<code>rejectRequests [].date</code>	Дата и время создания или последнего изменения состояния операции возврата, в формате уууу-ММ dd'T'HH:mm:ss	Да
<code>rejectRequests [].status</code>	Статус обработки операции возврата. Может принимать следующие значения: <ul style="list-style-type: none">• PROCESSING — заявка на отмену создана и обрабатывается;• SUCCESS — заявка на отмену успешно обработана, статус основного заказа изменен;	Да

	<ul style="list-style-type: none">• ERROR — заявка на отмену завершилась ошибкой, статус основного заказа остался "SUCCESS".	
--	---	--

Пример запроса и ответа приведены в приложении Е. Пример запроса и ответа для метода запроса заявки на отмену платежа.

5.5 УВЕДОМЛЕНИЕ ПАРТНЕРА О СТАТУСЕ ОПЕРАЦИИ

В системе предусмотрен функционал callback-нотификации мастер мерчанта о переходе заказа в финальный статус (успех или ошибка). Функционал реализован с использованием технологии отправки webhook-оповещений.

Структура webhook-оповещения:

```
{
  "id" : string,
  "transaction" : 21799382,
  "status" : string,
  "errorDetail" : null,
  "error" : 0,
  "message" : null,
  "bankPayment" : null,
  "date" : string,
  "attribute" : [ {
    "code" : string,
    "value" : string,
    "title" : null,
    "valueTitle" : null,
    "flags" : null
  } ],
  "menu" : {
    "service" : 8,
    "serviceName" : string,
    "category" : string
  },
  "sumOutcome" : 10000
}
```

}

Таблица 5.5.1 — Атрибуты webhook-оповещения

Параметр	Описание	Обяз.
id	Уникальный идентификатор заказа на стороне мастер мерчанта	Да
transaction	Уникальный идентификатор транзакции системы	Да
status	Статус операции, может принимать значения: <ul style="list-style-type: none">• SUCCESS — Успех• ERROR — Ошибка	Да
sumOutcome	Сумма заказа, в минорной единице валюты (копейки, центы и т.д.)	Да
error	Код ошибки	Нет
errorDetail	Детализирующий код ошибки	Нет
message	Описание ошибки	Нет
bankPayment	Объект данных о банковской транзакции. Описание объекта приведено в таблице 5.5.2	Нет
date	Дата и время транзакции, в формате уууу-ММ dd'T'HH:mm:ss	Нет
attribute	Объект данных атрибутов основной операции. Описание объекта приведено в таблице 5.5.3	Нет
menu	Объект данных сервиса. Описание объекта приведено	Нет

	в таблице 5.5.4	
--	-----------------	--

Таблица 5.5.2— Атрибуты объекта **bankPayment**

Параметр	Описание	Обяз.
status	Статус операции, может принимать значения: <ul style="list-style-type: none">• SUCCESS — Успех• ERROR — Ошибка	Нет
transaction	Уникальный идентификатор транзакции на стороне банка	Нет
attributes	Атрибуты банковской операции	Нет

Таблица 5.5.3— Атрибуты объекта **attribute**

Параметр	Описание	Обяз.
code	Код атрибута	Нет
value	Значение	Нет
title	Заголовок	Нет
valueTitle	Отображаемый заголовок	Нет
flags	Флаг	Нет

Таблица 5.5.4— Атрибуты объекта **menu**

Параметр	Описание	Обяз.
service	Код сервиса	Нет
serviceName	Название сервиса	Нет
category	Категория сервиса	Нет

В ответ от мерчанта системой будет ожидаться HTTP-ответ:

- **200** — webhook-оповещение успешно принято, система прекратит отправку;
- **401, 403** — доступ запрещен, система прекратит отправку;
- **404** — URL не найден, система прекратит отправку;
- **любой код, отличный от 200, 401, 403 и 404** — ошибка, система повторит отправку позже. Webhook-оповещение будет находиться в очереди на отправку 24 часа и будет периодически отправляться повторно. Если по истечению 24 часов, от мерчанта не будет получен успешный ответ, то система прекратит отправку.

Пример уведомления приведен в приложении F. Пример webhook-оповещения.

Для авторизации запросов webhook-оповещений используется проверка электронной подписи. В запросе передается HTTP-заголовок **X-Sign**, содержащий подпись webhook-сообщения.

Электронная подпись формируется от строки **<REQUEST METHOD>+<REQUEST FULL URI>+<REQUEST BODY>** по алгоритму SHA256withRSA в формате Base64 (UTF8).

Пример:

Webhook-оповещение отправляется мастер мерчанту по адресу —

`https://{host}:{port}/merchant/webhook`

Параметры URL (host, port) для отправки webhook-оповещений определяет мастер мерчант. Значение должно быть указано в настройках системы PSP через кабинет пользователя.

Подпись в данном примере будет иметь вид:

```
signBase64 (POST/merchant/webhook{"body":"body"})
```

Проверка подписи осуществляется по публичной части ключа, полученной от системы.

Пример генерации и проверки подписи приведен в приложении [А. Примеры генерации и проверки ЭП на java.](#)

6 ПРИЛОЖЕНИЕ

А. ПРИМЕРЫ ГЕНЕРАЦИИ И ПРОВЕРКИ ЭП НА JAVA

Пример генерации и проверки подписи для методов API:

```
import org.apache.commons.codec.binary.Base64;
import org.apache.commons.codec.binary.Hex;
import org.bouncycastle.openssl.PEMReader;
import org.bouncycastle.openssl.PasswordFinder;
import java.io.FileReader;
import java.io.IOException;
import java.security.*;

public class Signer {

    public static final String defaultAlgorithm = "SHA256withRSA";
    public static final String defaultCharset = "UTF-8";
    private final PrivateKey privateKey;
    private final PublicKey publicKey;

    public Signer(String publicKeyPath, String privateKeyPath) throws IOException
    {
        try (FileReader fr = new FileReader(publicKeyPath); PEMReader pr = new
        PEMReader(fr)) {
            publicKey = (PublicKey) pr.readObject();
        }
        try (FileReader fr = new FileReader(privateKeyPath); PEMReader pr = new
        PEMReader(fr)) {
            KeyPair kp = (KeyPair) pr.readObject();
            privateKey = kp.getPrivate();
        }
    }

    public boolean verifyBase64(String message, String signature) throws
    SignatureException {
        try {
            Signature sign = Signature.getInstance(defaultAlgorithm);
            sign.initVerify(publicKey);
            sign.update(message.getBytes(defaultCharset));
            return
            sign.verify(Base64.decodeBase64(signature.getBytes(defaultCharset)));
        } catch (Exception ex) {
            throw new SignatureException(ex);
        }
    }
}
```

```
    }  
}  
  
public String signBase64(String message) throws SignatureException {  
    try {  
        Signature sign = Signature.getInstance(defaultAlgorithm);  
        sign.initSign(privateKey);  
        sign.update(message.getBytes(defaultCharset));  
        return new String(Base64.encodeBase64(sign.sign()), defaultCharset);  
    } catch (Exception ex) {  
        throw new SignatureException(ex);  
    }  
}
```

Пример генерации и проверки подписи для webhook оповещений:

```
import org.apache.commons.codec.binary.Base64;  
import org.apache.commons.codec.binary.Hex;  
import org.bouncycastle.openssl.PEMReader;  
import org.bouncycastle.openssl.PasswordFinder;  
import java.io.FileReader;  
import java.io.IOException;  
import java.security.*;  
public class Signer {  
    public static final String defaultAlgorithm = "SHA256withRSA";  
    public static final String defaultCharset = "UTF-8";  
    private final PublicKey publicKey;  
    public Signer(String publicKeyPath) throws IOException {  
        try (FileReader fr = new FileReader(publicKeyPath);  
            PEMReader pr = new PEMReader(fr)) {  
            publicKey = (PublicKey) pr.readObject();  
        }  
    }  
    public boolean verifyBase64(String message, String signature) throws  
SignatureException {  
        try {  
            Signature sign = Signature.getInstance(defaultAlgorithm);  
            sign.initVerify(publicKey);  
            sign.update(message.getBytes(defaultCharset));  
            return  
sign.verify(Base64.decodeBase64(signature.getBytes(defaultCharset)));  
        } catch (Exception ex) {  
            throw new SignatureException(ex);  
        }  
    }  
}
```

```
}
```

В. ПРИМЕР ЗАПРОСА И ОТВЕТА ДЛЯ МЕТОДА ЗАПРОСА ИНФОРМАЦИИ ИЗ QR-КОДА

Пример запроса:

```
{
  "OperationTime": "2026-03-02T06:49:10.114+05:00",
  "QrData": "https://pay.tezqr.uz/p?
d=D003100D109100100132D22054686520736561726368D305100D403860D503347D61420
250328122426D732DDDC203F4B3FB0C8D669241D9AFDFD78"
}
```

Пример успешного ответа:

```
{
  "status": "SUCCESS",
  "errorCode": "000000",
  "errorMessage": null,
  "data": {
    "id": "external-id-123321",
    "merchantName": "The search",
    "merchantID": "12858",
    "country": "860",
    "city": "Ташкент",
    "merchantAddress": "улица Название, д. 5 ",
    "mcc": "7399",
    "terminalId": "100100132",
    "amount": 100,
    "currency": "860"
  }
}
```

С. ПРИМЕР ЗАПРОСА И ОТВЕТА ДЛЯ МЕТОДА ЗАПРОСА СТАТУСА ЗАКАЗА

Пример запроса статуса заказа:

```
/psp/external/api/payment/status?id=external-id-123321
```

Пример ответа на запрос статуса заказа:

```
{
  "id": "external-id-123321",
  "transaction": 1391191,
  "status": "SUCCESS",
  "errorDetail": null,
  "error": 0,
  "message": null,
  "bankPayment": {
    "status": "SUCCESS",
    "transaction": "ps_1027240831",
    "attributes": null,
    "error": null,
    "errorMessage": null
  },
  "date": "2025-05-18T07:48:37.264+0000",
  "attributes": [
    {
      "code": "QrData",
      "title": null,
      "value": "https://pay.tezqr.uz/p?
d=D003100D109100100132D22054686520736561726368D305100D403860D503347D61420
250328122426D732DDDC203F4B3FB0C8D669241D9AFDFD78",
      "valueTitle": null,
      "flags": 0
    }
  ],
  "menu": {
    "service": 5,
    "serviceName": "Оплата по QR",
    "category": "Оплата по QR"
  }
}
```

D. ПРИМЕР ЗАПРОСА И ОТВЕТА ДЛЯ ОПЕРАЦИИ ОТМЕНЫ ПЛАТЕЖА И ВОЗВРАТА СРЕДСТВ**Пример запроса:**

```
{
  "id": "86b82362-4ddb-39b0-90ba-947430c95ba4",
  "comment": "cancel",
  "sum": 2000,
  "currency": "UZS"
}
```

Пример ответа:

```
{
  "id": "86b82362-4ddb-39b0-90ba-947430c95ba4",
  "transaction": 1030562,
  "sum": 2000,
  "currency": "UZS",
  "comment": "cancel",
  "date": "2026-03-20T05:21:33.866+0000",
  "status": "PROCESSING"
}
```

E. ПРИМЕР ЗАПРОСА И ОТВЕТА ДЛЯ МЕТОДА ЗАПРОСА ЗАЯВКИ НА ОТМЕНУ ПЛАТЕЖА**Пример запроса:**

```
{
  "id": "86b82362-4ddb-39b0-90ba-947430c95ba4"
}
```

Пример ответа:

```
{
  "currency": "UZS",
  "error": 0,
  "errorMessage": null,
  "id": "86b82362-4ddb-39b0-90ba-947430c95ba4",
}
```

```
"rejectRequests": [  
  {  
    "comment": "cancel",  
    "currency": "UZS",  
    "date": "2026-03-20T05:21:33.866+0000",  
    "requestId": 6,  
    "status": "SUCCESS",  
    "sum": 2000  
  }  
],  
"status": "SUCCESS",  
"sum": 560000,  
"transaction": 1030562  
}
```

Ф. ПРИМЕР ВЕБНООК-ОПОВЕЩЕНИЯ

```
{  
  "id" : "external-id-123321",  
  "transaction" : 1391191,  
  "status" : "SUCCESS",  
  "errorDetail" : null,  
  "error" : 0,  
  "message" : null,  
  "bankPayment" : {  
    "status" : "SUCCESS",  
    "transaction" : "ps_13755949",  
    "attributes" : null  
  },  
  "date" : "2025-05-18",  
  "attribute" : [ {  
    "code" : "QrData",  
    "value" : "https://pay.tezqr.uz/p?  
d=D003100D109100100132D22054686520736561726368D305100D403860D503347D61420  
250328122426D732DDDC203F4B3FB0C8D669241D9AFDFD78",  
    "title" : null,  
    "valueTitle" : null,  
    "flags" : null  
  } ],  
  "menu" : {  

```

```
"service" : 5,  
"serviceName" : "Оплата по QR",  
"category" : "Оплата по QR"  
},  
"sumOutcome" : 24796  
}
```

G. СТАТУСЫ ОПЕРАЦИЙ

Таблица G.1 — Статусы заказа

Статус	Описание	Финальность
SUCCESS	Операция успешно обработана	Да
ERROR	Ошибка проведения операции	Да
NOT_FOUND	Не найден id операции	Да
PROCESSING	Проведение	Нет
VALIDATION_ERROR	Ошибка валидации	Нет

Таблица G.2 — Возможные значения параметра **errorCode**

Код	Описание
000000	Успешная обработка
010000	Общий отказ
020000	ТСТ не найдено в системе
030000	Оплата в ТСТ приостановлена
040000	Операция не разрешена Банку партнеру

050000	Некорректный формат запроса
060000	Подозрительная операция
070000	Сумма возврата в рамках заказа больше суммы оригинальной
080000	Нарушена последовательность запросов
080001	Заказ оплачен другим банком
090000	Оригинальный заказ для возврата не найден
100000	Указанный заказ не найден
110000	Недопустимая сумма в заказе
120000	Партнер (MemberId) не найден в системе
130000	Операция не была проведена успешно
170000	Некорректный settleDate
180000	Неверный порядковый номер страницы