

# **Платёжный шлюз «ВсеПлатежи»**

**Переводы на карты физических лиц**

## Оглавление

История изменений.....	3
Раздел 1 Определения .....	4
Раздел 2 Общие сведения .....	5
2.1 Защищённое соединение .....	5
2.2 Характеристика API.....	5
2.3 Способы выполнения переводов на карты физических лиц.....	5
2.3.1 Перевод денежных средств с участием Получателя.....	5
2.3.2 Перевод денежных средств с предварительной регистрацией карты Получателя .....	6
2.3.3 Уведомление об успешном проведении операции .....	6
Раздел 3 API .....	8
3.1 Адреса серверов .....	8
3.2 Поддерживаемые запросы.....	8
3.2.1 Запрос на проведение перевода с участием Получателя.....	9
3.2.2 Запрос на регистрацию карты .....	10
3.2.3 Запрос на перевод без участия Получателя .....	10
3.2.4 Запрос статуса регистрации карты .....	12
3.2.5 Запрос статуса перевода .....	12
3.2.6 Коды и тексты состояний .....	13
3.3 Алгоритм формирования НМАС.....	14
3.3.1 Подготовка строки данных для НМАС.....	14
3.3.2 Генерация НМАС .....	15
3.4 Примеры для некоторых языков программирования .....	16
3.4.1 PHP.....	16
3.4.2 Java.....	16
Раздел 4 Реестр проведённых операций.....	17
4.1 Формирование и отправка реестра .....	17
4.2 Формат письма .....	17
4.3 Формат файла реестра .....	17
Раздел 5 Коды ответов .....	19
5.1 Общие принципы .....	19
5.2 Коды ответов* .....	19

## История изменений

Версия	Изменения
1.0	Создан документ

## Раздел 1 Определения

---

**Платёжный шлюз «ВсеПлатежи»** — совокупность программных и аппаратных средств, выполняющих: 1) обработку запросов Партнера на проведение операций, 2) проведение операций, 3) передачу результатов проведения операций Партнерам. Далее используется термин «Платёжный шлюз».

**Партнер** — клиент Платёжного шлюза, с которым заключён договор на оказание услуг по переводу денежных средств на карты физических лиц и налажено техническое взаимодействие.

**Получатель** — физическое лицо, на карту которого переводятся денежные средства. Получатель не является клиентом Платёжного шлюза.

**Банк-эмитент** — банк, выпустивший в обращение карту Получателя.

**API** — программный интерфейс для взаимодействия с каким-либо приложением или системой, в частности, с сервером Платёжного шлюза.

**Общий секретный ключ** — набор случайных цифр в шестнадцатеричном формате, сгенерированный модулем безопасности Платёжного шлюза для формирования подписи (HMAC). Ключ присваивается терминалу Партнера и должен храниться в тайне у обеих сторон.

**HMAC** — *hash-based message authenticate code* (код аутентификации сообщений) — набор символов, сформированный при обработке входящих параметров по алгоритму SHA256 с использованием общего секретного ключа. HMAC передаётся отдельным параметром `sign` в запросах от Партнера к API Платёжного шлюза и передаче ответов на эти запросы обратно Партнеру. Предназначен для обеспечения целостности запроса и обоюдной аутентификации Платёжного шлюза и Партнера.

## Раздел 2 Общие сведения

---

### 2.1 Защищённое соединение

Все взаимодействия с платёжным шлюзом производятся по протоколу HTTPS. Для защиты передаваемой информации используется протокол TLS версии 1.2. Протоколы SSL всех версий и TLS версий ниже 1.2 не поддерживаются.

### 2.2 Характеристика API

API Платежного шлюза базируется на технологии REST API. В системе используются ресурсно-ориентированные точки входа (endpoints), а все ответы, возвращаемые Платежным шлюзом, имеют формат JSON.

API является асинхронным (незначительная часть запросов работает синхронно).

### 2.3 Способы выполнения переводов на карты физических лиц

Доступно два способа перевода на карты физических лиц:

- С участием Получателя перевода;
- Без участия Получателя с предварительной регистрацией карты на Платежном шлюзе.

#### 2.3.1 Перевод денежных средств с участием Получателя

Получатель инициирует получение перевода на стороне Партнера (например, нажатием на кнопку «Вывести средства на карту»). Партнер формирует запрос с подписью и отправляет его как HTML-форму с непосредственным перенаправлением Получателя на сайт Платёжного шлюза. При данном запросе шлюз проверяет параметры запроса и отображает форму для ввода номера карты. Если запрос от Партнера сформирован неверно, то Получатель увидит страницу с ошибкой «Операция отклонена».

После успешной верификации введённых Получателем данных выполняются следующие шаги:

1. В случае успешного проведения операции Платёжный шлюз асинхронно (без блокировки пользовательского интерфейса) отправляет Партнеру уведомление (см. п. 2.3.3).
2. Получатель перенаправляется на страницу сайта Партнера по адресу, указанному в `clientBackUrl`, с передачей дополнительного параметра `result=0`, говорящего об успехе операции перевода. В случае неудачной отправки уведомления оно будет повторно отправляться заданное в конфигурации Платёжного шлюза число раз (по умолчанию – 3 раза) через заданный таймаут (по умолчанию – 2 минуты).

3. В случае отказа в проведении перевода Получателю (недостаточно средств на балансе Партнера или технические проблемы) отображается страница, где описана причина отказа (при возможности ее определения). В этом случае Получатель может вернуться на сайт Партнера (и еще раз попробовать повторить операцию), кликнув соответствующую ссылку на форме оплаты (Получатель будет перенаправлен на `clientBackUrl` с передачей дополнительного параметра `result`, **не равного 0**, говорящего об отказе от операции оплаты).

### 2.3.2 Перевод денежных средств с предварительной регистрацией карты Получателя

На стороне Партнера должна быть предусмотрена функция регистрации карт Получателя на Платежном шлюзе (например, при регистрации Получателя на сайте Партнера может быть шаг привязки карт для дальнейших выплат).

При запуске процесса регистрации карты Партнер формирует запрос аналогично процессу, описанному в разделе 2.3.1. Запрос отправляется на URL для регистрации карты. В результате операции Партнер в асинхронном режиме получит идентификатор карты. Шаги аналогичны шагам, описанным в разделе 2.3.1 (производится только регистрация карты без выполнения операции перевода. При регистрации карты НЕ происходит обращения к банку-эмитенту для проверки наличия такой карты).

При наступлении необходимости перевода на карту получателя Партнер формирует запрос с идентификатором карты Получателя и суммы перевода. В асинхронном режиме Партнеру передается уведомление об успешном переводе.

Информацию о статусе перевода можно получить путем запроса статуса перевода, если не получено уведомление об успешном переводе в течение 15 минут с момента отправки запроса на перевод.

### 2.3.3 Уведомление об успешном проведении операции

После успешной операции Платёжный шлюз асинхронно отправляет уведомление по протоколу HTTP в виде POST-запроса (с заголовком `Content-Type: application/x-www-form-urlencoded`) на согласованный заранее с Партнером URL (есть возможность настройки разных URL для каждого типа уведомлений).

**Операция регистрации карты. Список отправляемых параметров:**

1. `cardCheckinRequestNumber`
2. `merchant`
3. `terminal`
4. `cardNumber` (в маскированном виде 6\*4)
5. `cardId`

Для дальнейших переводов без участия Получателя используется параметр `cardId`.

Подпись генерируется для указанных параметров по алгоритму, описанному в п. 3.3, и, в целях безопасности, должна проверяться на стороне Партнера.

**Операция перевода на карту. Список отправляемых параметров:**

1. orderId
2. transactionId
3. transactionDateTime
4. terminal
5. merchant
6. orderDateTime
7. cardNumber
8. cardId
9. amount
10. sign

Параметр cardId возвращается всегда, независимо от типа перевода (с участием или без участия Получателя), т.е. при переводе с участием Получателя происходит регистрация карты, который можно использовать для дальнейших переводов без участия Получателя.

Подпись генерируется для указанных параметров по алгоритму, описанному в п. 3.3, и, в целях безопасности, должна проверяться на стороне Партнера

## Раздел 3 API

### 3.1 Адреса серверов

Данные адреса необходимо использовать в качестве базовых, прибавляя к ним относительные URL, описанные далее.

Основной	Сообщается Партнеру при выдаче доступа
Тестовый	Сообщается Партнеру при выдаче доступа

### 3.2 Поддерживаемые запросы

В данной таблице в общем виде описаны поддерживаемые методы

Запрос	HTTP-метод и относительный URL	Результат
Запрос на проведение перевода с участием Получателя	POST /moneytransfer	<p><i>Успешно:</i> Отображение страницы с формой оплаты (формой ввода данных карты).</p> <p><i>Ошибка:</i> Отображение страницы с описанием ошибки и HTTP-кодом, соответствующим типу ошибки.</p>
Запрос на регистрацию карты Получателя	POST /checkincardpage	<p><i>Успешно:</i> Отображение страницы с формой регистрации карты (формой ввода данных карты).</p> <p><i>Ошибка:</i> Пустой ответ с HTTP-кодом, соответствующим типу ошибки.</p>
Запрос на проведение перевода на ранее зарегистрированную карту (без участия Получателя)	POST /transferbycardid	<p><i>Успешно:</i> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><i>Ошибка:</i> Ответ с соответствующим HTTP- кодом</p>



		и JSON-строкой в теле сообщения.
Запрос статуса перевода	POST /transerstatus	<p><b>Успешно:</b> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><b>Ошибка:</b> Пустой ответ с HTTP-кодом, соответствующим типу ошибки.</p>
Запрос статуса регистрации карты	POST /checkinstatus	<p><b>Успешно:</b> Ответ с HTTP- кодом 200 и JSON-строкой в теле сообщения.</p> <p><b>Ошибка:</b> Пустой ответ с HTTP-кодом, соответствующим типу ошибки.</p>

### 3.2.1 Запрос на проведение перевода с участием Получателя

#### Базовые параметры

URL	/moneytransfer
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

#### Параметры запроса

Название	Описание	Ограничения, формат, длина
orderId	Номер перевода	Уникальный для терминала, Символьный, 1-50
amount	Сумма перевода в рублях	Символьный, с двумя знаками после точки, > 0.00
merchant	Идентификатор Партнера	Числовой, 1-50
terminal	Номер терминала Партнера	Числовой, 1-50
clientBackUrl	URL для возврата на сайт Партнера	Символьный, 1-255
description	Описание перевода	Символьный, 1-255 <b>Опциональный</b>
email	Адрес электронной почты Получателя	Символьный, формат: [a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+ 1-255 <b>Опциональный</b>
phone	Номер телефона Получателя	Символьный формат [0-9]{10} 10 символов

		<b>Оptionальный</b>
sign	Подпись запроса	Символьный, 64 (для HMAC SHA256)

### 3.2.2 Запрос на регистрацию карты

#### Базовые параметры

URL	/checkincardpage
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

#### Параметры запроса

Название	Описание	Ограничения, формат, длина
cardCheckinRequestNumber	Номер операции регистрации	Уникальный для терминала, числовой, 1-50
merchant	Идентификатор Партнера	Числовой, 1-50
terminal	Номер терминала Партнера	Числовой, 1-50
clientBackUrl	URL для возврата на сайт Партнера	Символьный, 1-255
email	Адрес электронной почты Получателя	Символьный, формат: [a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+ 1-255 <b>Оptionальный</b>
phone	Номер телефона Получателя	Символьный формат [0-9]{10} 10 символов <b>Оptionальный</b>
sign	Подпись запроса	Символьный, 64 (для HMAC SHA256)

### 3.2.3 Запрос на перевод без участия Получателя

#### Базовые параметры

URL	/transferbycardid
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

#### Параметры запроса

Запрос на перевод без участия Получателя карты содержит те же параметры, что и запрос на

проведение перевода с участием Получателя (п. 3.2.1) и обязательный параметр, содержащий идентификатор:

Название	Описание	Ограничения, формат, длина
orderId	Номер перевода	Уникальный для терминала, числовой, 1-50
cardId	Идентификатор карты получателя в базе Платежного шлюза	Строковый, 1-50
amount	Сумма перевода в рублях	Числовой, с двумя знаками после точки, > 0.00
merchant	Идентификатор Партнера	Числовой, 1-50
terminal	Номер терминала Партнера	Числовой, 1-50
description	Описание перевода	Символьный, 1-255 <b>Опциональный</b>
email	Адрес электронной почты Получателя	Символьный, формат: [a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+ 1-255 <b>Опциональный</b>
phone	Номер телефона Получателя	Символьный формат [0-9]{10} 10 символов <b>Опциональный</b>
sign	Подпись запроса	Символьный, 64 (для HMAC SHA256)

Примеры ответов:

```
{"paramsMap":
  {"rc":"500"}
}
```

```
{"paramsMap":
  {"amount":"10.01",
   "description":"Перевод на карту по id",
   "orderId":"1569575702249",
   "rc":"214",
   "sign":"409bb09a3d7991efc6103c6ef222bb1831651177ffb1a5f0f69f8af0de4de489"}
}
```

```
{"paramsMap":
  {"amount":"10.01",
   "description":"Перевод на карту по id",
   "merchant":"20000001",
   "orderId":"1569577973275",
   "rc":"58",
   "sign":"9379913f3dcba08411d494ab537b9e0cf58e451ac60a067f1c45a1edad80d2f1",
   "terminal":"200000011"}
}
```

Коды ответов rc описаны в Раздел 5.

### 3.2.4 Запрос статуса регистрации карты

#### Базовые параметры

URL	/checkinstatus
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

#### Параметры запроса

Название	Описание
cardCheckinRequestNumber	Номер операции
merchant	Идентификатор Партнера
terminal	Номер терминала Партнера
sign	HMAC запроса

Формат значений параметров смотрите в п. 3.2.1

#### Успешный ответ содержит JSON-структуру

```
{
  "data": {
    "checkinRequestNumber": "<номер операции>",
    "cardId": "<идентификатор>",
    "terminal": "<номер терминала Партнера>",
    "merchant": "<идентификатор Партнера>",
    "orderStatusCode": "1",
    "orderStatusText": "Обработан"
  }
}
```

Статусы описаны в п. 3.2.6

### 3.2.5 Запрос статуса перевода

#### Базовые параметры

URL	/transferstatus
Тип запроса	POST
Требуемые HTTP заголовки	Content-Type: application/x-www-form-urlencoded

#### Параметры запроса

Название	Описание
orderId	Номер перевода
merchant	Идентификатор Партнера

terminal	Номер терминала Партнера
sign	HMAC запроса

Формат значений параметров смотрите в п. 3.2.1

### Успешный ответ содержит JSON-структуру

```
{
  "data": {
    "transactions": [
      {
        "dateTime": "2019-08-14 13:11:58.864",
        "cardNumber": "220003*****1855",
        "amount": "10.11",
        "transactionId": "123467",
        "cardId": "5144"
      }
    ],
    "orderStatusText": "Подтвержден",
    "amount": "10.01",
    "terminal": "200000011",
    "merchant": "20000001",
    "orderId": "1565766710943",
    "orderStatusCode": "2"
  }
}
```

Статусы описаны в п. 3.2.6

## 3.2.6 Коды и тексты состояний

### Операция перевода

orderStatusCode	orderStatusText	Описание
0	Создан	Получен запрос от Партнёра на перевод
1	В обработке	Получатель ввел номер карты, запущен процесс обработки информации
2	Зачислен	Перевод проведен успешно (информация отправлена в банк-эмитент)
3	Подтвержден	Партнеру успешно отправлено уведомление об успешном переводе
4	Просрочен	Перевод не был проведен за отведенное время (например, получатель закрыл страницу без ввода номера карты для зачисления денежных средств)
5	Отклонен	Перевод был отклонен (по разным причинам – отказ банка, недостаток баланса Партнера)

**Операция регистрации**

<b>orderStatusCode</b>	<b>orderStatusText</b>	<b>Описание</b>
0	Создан	Получен запрос от Партнёра на регистрацию карты
1	Обработан	Карта успешно зарегистрирована
4	Просрочен	Регистрация не была проведена за отведённое время (например, получатель закрыл страницу без ввода номера карты для зачисления денежных средств)
5	Отклонен	Регистрация была отклонена по какой-либо ошибке

**Возвращаемые ошибки**

<b>Код HTTP</b>	<b>Причина</b>
235 Token not found	В базе не найден идентификатор карты
400 Bad Request	Неверно указаны параметры запроса
401 Unauthorized	Запрос не аутентифицирован (неверная подпись НМАС)
404 Not Found	Запрос не найден по указанным параметрам

**3.3 Алгоритм формирования НМАС**

НМАС должен проверяться Платёжным шлюзом при получении запроса на проведение операции от Партнера и при получении ответа по результату проведения операции от Платёжного шлюза Партнером. НМАС формируется в два этапа: 1 – формирование строки данных для НМАС, 2 – формирование НМАС.

**3.3.1 Подготовка строки данных для НМАС****Этапы:**

1. Каждое значение параметра дополняется его длиной: «длина текстового значения параметра в байтах» + «значение» (например, значение 1000.00 для использования в подписи будет иметь вид 71000.00, <https://vp.ru> → 13<https://vp.ru>, а оплата услуги → 25оплата услуги, так как русские буквы в кодировке UTF-8 занимают по два байта, плюс пробел – один байт).
2. Имена параметров должны быть отсортированы в алфавитном порядке.
3. Значения параметров, полученные в п.1, соединяются в одну строку без разделителей в порядке следования отсортированных именованных.

**Важно:**

- Не нужно выполнять кодирование URL (encoding) или экранирование HTML-символов (escaping). Например, знак & не должен преобразовываться в &amp;.
- Для значений параметров должна использоваться кодировка UTF-8.
- Параметр `sign` на этапе подготовки строки для подписи не используется.

**Пример:**

Допустим, Получатель инициировал процесс перевода и Партнер имеет следующие значения для запроса с целью перевода денежных средств на карту Получателя:

```
orderId=10000000001
amount=100.00
merchant=777
terminal=1001
clientBackUrl=https://example-merchant:8081/back-from-pay
description=Оплата по договору
```

По условиям, чтобы подготовить значения параметров для подписи, необходимо выстроить их по названию параметров:

```
amount=100.00
clientBackUrl=https://example-merchant:8081/back-from-pay
description=Оплата по договору
merchant=777
orderId=10000000001
terminal=1001
```

Затем значения параметров преобразуются и складываются:

```
6100.0043https://example-merchant:8081/back-from-pay34Оплата по
договору3777111000000000141001
```

Строка готова для генерации подписи.

**3.3.2 Генерация HMAC**

Для формирования HMAC используется алгоритм «HMACSHA256». В качестве ключа используется секретный ключ терминала Партнер. Ключ представляет собой последовательность байт в HEX-формате, например:

```
b22ec899aaf398624c14305d56a3aa98095523fe
```

Но для формирования HMAC он должен быть преобразован в бинарный массив. Данный ключ для наглядности можно представить в десятичном формате таким образом:

```
[178, 46, 200, 153, 170, 243, 152, 98, 76, 20, 48, 93, 86, 163, 170, 152, 9, 85, 35, 255]
```

Подпись HMAC для строки из предыдущего примера с использованием данного ключа имеет вид:

```
5d3973c71f2fc12e8b1ff91dad63b58c7e377cccbcd6bf01d3621ab3bd44189d.
```

## 3.4 Примеры для некоторых языков программирования

### 3.4.1 PHP

```
$stringToSign = '510.0144https://example-merchant:8081/pay-  
result=200340плата по договору3777111000000000141001';  
$shared_key = 'b22ec899aaf398624c14305d56a3aa98095523ff';  
$hmac = hash_hmac('SHA256', $stringToSign, pack('H*', $shared_key));
```

Результат в `$hmac`:

```
79c1947a8a9fced811af0a2f357aebdf027256761b926866eac65b4652323bcb
```

### 3.4.2 Java

```
import org.apache.commons.codec.binary.Hex;  
import javax.crypto.Mac;  
import javax.crypto.spec.SecretKeySpec;  
import java.nio.charset.Charset;  
import java.nio.charset.StandardCharsets;  
import java.security.MessageDigest;  
...  
String stringToSign = '510.0144https://example-merchant:8081/pay-  
result=200340плата по договору 3777111000000000141001';  
Mac mac = Mac.getInstance("HmacSHA256");  
byte[] signingKeyBytes = Hex.decodeHex(sharedSecretKey.toCharArray());  
SecretKeySpec signingKey =  
    new SecretKeySpec(signingKeyBytes, mac.getAlgorithm());  
mac.init(signingKey);  
byte[] rawHmac = mac.doFinal(stringToSign.getBytes(StandardCharsets.UTF_8));  
byte[] hexBytes = new Hex().encode(rawHmac);  
String stringHmac = new String(hexBytes, StandardCharsets.UTF_8);  
return stringHmac
```



## Раздел 4 Реестр проведенных операций

### 4.1 Формирование и отправка реестра

Реестр проведенных операций формируется на следующие сутки и отправляется вложением в отдельном письме на адрес электронной почты, предоставленный Партнером для этих целей.

### 4.2 Формат письма

**Тема:**

*Реестр переводов за ДД.ММ.ГГГГ г. <наименование Партнера>*

**Содержание:**

*Реестр переводов за ДД.ММ.ГГГГ г. <наименование Партнера>*

*Количество записей в реестре: <фактическое количество записей в реестре>*

**Вложение:**

Файл реестра.

### 4.3 Формат файла реестра

Реестр формируется в виде **.xlsx**-файла (в формате *Excel 2007* и выше).

**Наименование файла:**

*MERCH\_ID\_TERM\_ID\_ДД.ММ.ГГГГ.xlsx*, где

- MERCH\_ID – номер Партнера;
- TERM\_ID – номер терминала;
- ДД.ММ.ГГГГ – дата, за которую сформирован реестр.

**Структура файла:**

Наименование поля	Описание
Номер перевода	Номер перевода. <i>Формат: текстовый.</i>
Код авторизации	Код авторизации операции. <i>Формат: текстовый.</i>

Дата и время операции	Дата и время перехода соответствующей транзакции в статус «Зачислена». <i>Формат: дата и время (ДД.ММ.ГГГГ чч:мм:сс).</i>
Номер карты	Маскированный номер карты (6*4). <i>Формат: текстовый.</i>
Сумма операции	Общая сумма заказа или возврата. Значение всегда положительное, даже в случае возврата. <i>Формат: числовой #0.00*.</i>
Комиссия с Партнера по данной операции	Комиссия шлюза за проведение операции. <i>Формат: числовой #0.00*.</i>

**Примечание:**

\* В зависимости от настроек операционной системы, суммы могут отображаться с точкой или с запятой в качестве десятичного разделителя. Для российской локали используется запятая, как показано в примере ниже.

## Раздел 5 Коды ответов

### 5.1 Общие принципы

Код ответа записывается в поле гс.

Коды ответа до 200 соответствуют ISO 8583. Если эквайеры используют собственные коды ответов, то данные коды соотносятся с ISO 8583 по мере возможности. В случае, если ответ эквайера нельзя соотнести с ISO 8583, код будет равен 501.

Коды ответов, начиная с 201, соответствуют ошибкам обработки запросов платежным шлюзом «ВсеПлатежи».

### 5.2 Коды ответов\*

Код	Расшифровка
0	Успешное проведение операции
201	Сумма меньше либо равна нулю
202	Сумма имеет неверный формат
203	Ссылка для возврата к мерчанту не указана
204	Ссылка для возврата к мерчанту имеет неверный формат
205	Email имеет неверный формат
206	Описание платежа имеет неверный формат
207	Идентификатор плательщика имеет неверный формат
208	Номер мерчанта или номер терминала имеет неверный формат
209	Номер платежа не указан
210	Номер платежа имеет неверный формат
211	Данный тип интеграции не поддерживается
212	Идентификатор сессии плательщика не указан
213	Терминал мерчанта или мерчант не найден
214	Платёж с таким номером уже существует
215	Платёж с таким номером не найден
216	Терминал мерчанта отключен
217	Средства не были заблокированы
218	В настоящее время уже выполняется списание средств
219	По данному платежу уже было выполнено списание средств
220	В настоящее время уже выполняется разблокировка средств
221	В настоящее время уже выполняется процесс оплаты

222	В настоящее время уже выполняется блокировка средств
223	Сумма не соответствует ожидаемой
224	Неверный номер карты
225	Карта просрочена
226	MD имеет неверный формат
227	Указан неверный MD
228	Результат с 3DS не ожидается
229	Операция не ожидается
230	Неверные данные карты
231	IP адрес клиента указан не верно
232	Невалидная подпись
234	Номер телефона имеет неверный формат
235	Токен карты не найден
500	Внутренняя ошибка
501	Ошибка на стороне эквайера
502	Необходимо пройти 3ds

\* Приведены все возможные коды. Не все из них могут приходить в ответах на запросы, описанные в данной документе.