

Visa QIWI Wallet

Протокол взаимодействия провайдеров с платежным сервисом Visa QIWI Wallet

Версия 2.1

Оглавление

1.	Основные сведения	2
2.	Интерфейс Visa QIWI Wallet	3
2.1.	Выставление счета пользователю	3
2.2.	Запрос статуса счета	5
2.3.	Переадресация для оплаты счета	6
2.4.	Отмена неоплаченного выставленного счета	7
2.5.	Возврат средств по оплаченному счету	9
2.6.	Проверка статуса возврата	10
2.7.	Ответ сервера	11
3.	Интерфейс провайдера	15
4.	Авторизация при запросах	17
4.1.	Авторизация при запросах на сервер Visa QIWI Wallet	17
4.2.	Авторизация при уведомлениях на сервер провайдера	17
5.	Стандартная схема работы	19
6.	Приложения	21
6.1.	Статусы счетов	21
6.2.	Статусы платежей	21
6.3.	Коды ошибок	22
6.4.	Коды завершения уведомлений	22

1. Основные сведения

Взаимодействие между сервером Visa QIWI Wallet и провайдеров происходит по HTTP-протоколу.

Данные при запросах в сторону Visa QIWI Wallet передаются в формате параметров HTTP-запроса. В ответ данные возвращаются в одном из двух форматов в соответствии со значением заголовка “Accept”, передаваемого в запросе:

- XML (значения заголовка “Accept”: “application/xml”, “text/xml”);
- JSON (значения заголовка “Accept”: “application/json”, “text/json”).

При запросах на сервер провайдера данные передаются в виде параметров HTTP-запроса. Ответ ожидается в XML-формате.

Для получения уведомлений провайдер должен принимать запросы из следующих подсетей:

- 91.232.230.0/23
- 79.142.16.0/20

Для обеспечения безопасности передачи данных все запросы в сторону сервера Visa QIWI Wallet шифруются с помощью SSL. HTTP-запросы по нешифрованному каналу не поддерживаются. При запросах на сервер провайдера также возможно использование SSL (возможно использование самоподписанных сертификатов) или простой подписи по схеме HMAC-SHA1 (подробнее см. в главе [“Авторизация”](#)). Провайдер должен проверять серверный сертификат Visa QIWI Wallet, используя стандартный алгоритм проверки сертификатов.

2. Интерфейс Visa QIWI Wallet

- [Выставление счета пользователю](#)
- [Получение статуса выставленного счета](#)
- [Отмена неоплаченного выставленного счета](#)
- [Возврат \(частичный или полный\) средств по оплаченному счету](#)
- [Проверка статуса возврата](#)

2.1. Выставление счета пользователю

Для выставления счета пользователю Visa QIWI Wallet от провайдера необходимо послать PUT-запрос по адресу:

https://w.qiwi.com/api/v2/prv/{prv_id}/bills/{bill_id}

- **{prv_id}** - числовой идентификатор провайдера
- **{bill_id}** - уникальный идентификатор счета на стороне провайдера

Параметры запроса:

Имя параметра	Формат значения	Regexp	Описание
user	Строка вида “ tel:{phone_number} ”, где {phone_number} – номер телефона в международном формате	$^{\wedge}\text{tel}:\backslash+\backslash\text{d}\{1,15\}\$$	Идентификатор кошелька пользователя, которому выставляется счет. Представляет собой номер телефона пользователя с префиксом “tel:”
amount	Положительное число, округленное до 2 или 3 знаков после запятой.	$^{\wedge}\backslash\text{d}+(\backslash.\backslash\text{d}\{0,3\})?\$$	Сумма, на которую выставляется счет. Способ округления зависит от валюты
ccy	Трёхбуквенная аббревиатура	$^{\wedge}[\text{a-zA-Z}]\{3\}\$$	Идентификатор валюты (Alpha-3 ISO 4217 код)
comment	Любой текст	$^{\wedge}\backslash.\{0,255\}\$$	Комментарий к счету


```
HTTP/1.1 200 OK
Content-Type: text/json
```

```
{"response": {
  "result_code": 0,
  "bill": {
    "bill_id": "BILL-1",
    "amount": "10.00",
    "ccy": "RUB",
    "status": "waiting",
    "error": 0,
    "user": "tel:+79031234567",
    "comment": "test"
  }
}}
```

2.2. *Запрос статуса счета*

Для запроса текущего статуса счета необходимо послать GET-запрос по адресу:

https://w.qiwi.com/api/v2/prv/{prv_id}/bills/{bill_id}

- **{prv_id}** - числовой идентификатор провайдера
- **{bill_id}** - уникальный идентификатор счета на стороне провайдера (любая непустая строка до 200 символов).

Параметров запроса нет.

В ответ возвращается код результата выполнения операции и, в случае успешного ее завершения, все данные о запрошенном счете (подробнее см. [ответ сервера](#)).

Пример:

```
GET /api/v2/prv/2042/bills/BILL-1
Accept: text/json
Authorization: Basic MjA0MjpwOZxN0Cg==
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

```
HTTP/1.1 200 OK
Content-Type: text/json
```

```
{"response": {
  "result_code": 0,
  "bill": {
    "bill_id": "BILL-1",
    "amount": "10.00",
    "ccy": "RUB",
    "status": "waiting",
    "error": 0,
    "user": "tel:+79031234567",
    "comment": "test"
  }
}}
```

2.3. Переадресация для оплаты счета

Провайдер должен предложить пользователю немедленно оплатить счет, переадресуя его на страницу Visa QIWI Wallet или открыв ее в iFrame по одному из следующих URL:

- <https://w.qiwi.com/order/external/main.action?shop=xxxx&transaction=yyyyyyyyy> – to redirect
- <https://w.qiwi.com/order/external/main.action?shop=xxxx&transaction=yyyyyyyyy&iframe=true> – to use iFrame

Эти ссылки работают одинаково за исключением того, что вторая отображает страницу в более компактном виде, удобном для встраивания ее в сайт провайдера

- Обе ссылки принимают следующие GET параметры:

Имя параметра	Тип	Описание	Пример
shop	строка	Идентификатор провайдера. Соответствует prv_id из запроса на выставление счета.	123
transaction	строка	Идентификатор счета на стороне провайдера. Соответствует параметру bill_id из запроса на выставление счета.	abcde12345
successUrl	URL-закодированная строка	URL для переадресации в случае успешного выставления счета	http%3A%2F%2Fmystore.com%2Fsuccess%3Fa%3D1%26b%3D2
failUrl	URL-закодированная строка	URL для переадресации в случае неуспешного выставления счета	http%3A%2F%2Fmystore.com%2Ffail%3Fa%3D1%26b%3D2

Visa QIWI Wallet переадресует пользователя после оплаты на successUrl или failUrl с дополнительным параметром order, в котором будет передан идентификатор счета, отправленный в параметре transaction из изначального запроса. Используя этот параметр, провайдер может отобразить необходимую информацию на своей стороне.

Пример:

- Провайдер переадресует на URL:
<https://w.giwi.com/order/external/main.action?shop=2042&transaction=123&successUrl=http://mystore.com/success?a=1&b=2&failUrl=http://mystore.com/fail?a=1&b=2>
- Клиент оплачивает счет удобным ему способом
- Visa QIWI Wallet переадресует клиента на URL
<http://mystore.com/success?a=1&b=2&order=1234567>.

ВНИМАНИЕ: Не основывайте логику успешности оплаты счета на факте редиректа на successUrl – это всего лишь удобное интерфейсное решение. Для принятия окончательного решения о предоставлении услуги или товара клиенту необходимо дождаться уведомления от сервера Visa QIWI Wallet.

2.4. Отмена неоплаченного выставленного счета

С помощью данного запроса провайдер может отменить ранее выставленный им счет, но только в том случае, если пользователь еще не инициировал его оплату. Для этого необходимо послать PATCH-запрос по адресу:

https://w.giwi.com/api/v2/prv/{prv_id}/bills/{bill_id}

- **{prv_id}** - числовой идентификатор провайдера
- **{bill_id}** - уникальный идентификатор счета на стороне провайдера (любая непустая строка до 200 символов).

Имя параметра	Формат значения	Regex	Описание
status	"rejected"	^rejected\$	Новый статус счета (отменен)

В ответ возвращается код результата выполнения операции и, в случае успешного ее завершения, все данные о счете (подробнее см. [ответ сервера](#)).

Пример:

```
PATCH /api/v2/prv/2042/bills/BILL-2
Accept: text/json
Authorization: Basic MjA0Mjp0ZXN0Cg==
Content-Type: application/x-www-form-urlencoded; charset=utf-8

status=rejected
```

```
HTTP/1.1 200 OK
Content-Type: text/json
```

```
{"response": {
  "result_code": 0,
  "bill": {
    "bill_id": "BILL-2",
    "amount": "10.00",
    "ccy": "RUB",
    "status": "rejected",
    "error": 0,
    "user": "tel:+79031234567",
    "comment": "test"
  }
}}
```

2.5. Возврат средств по оплаченному счету

С помощью данного запроса можно произвести полный или частичный (на неполную сумму счета) возврат средств по счету клиенту на его Visa QIWI Wallet. При этом сгенерируется платеж, обратный платежу на оплату счета. В случае если сумма, переданная в запросе, превышает сумму самого счета либо сумму счета, оставшуюся после предыдущих возвратов, будет произведен возврат на оставшуюся сумму. Для осуществления возврата необходимо послать PUT-запрос по адресу:

https://w.qiwi.com/api/v2/prv/{prv_id}/bills/{bill_id}/refund/{refund_id}

- **{prv_id}** - числовой идентификатор провайдера
- **{bill_id}** - уникальный идентификатор счета на стороне провайдера (любая непустая строка до 200 символов)
- **{refund_id}** - идентификатор отмены, уникальный в рамках отмен одного счета. Целое число от 1 до 999999999.

Параметры запроса:

Имя параметра	Формат значения	Regex	Описание
amount	Положительное число, округленное до 2 или 3 знаков после запятой.	<code>^\d+(\.\d{0,3})?\$\$</code>	Сумма возврата. Должна быть меньше либо равна сумме счета, по которому производится возврат. Способ округления зависит от валюты

В ответ возвращается результат выполнения операции и все данные об отмене (подробнее см. [ответ сервера](#)).

Пример:

```
PUT /api/v2/prv/2042/bills/BILL-1/refund/1
```

```
Accept: text/json
```

```
Authorization: Basic MjA0Mjp0ZXN0Cg==
```

```
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

```
amount=5.0
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/json
```

```
{
  "response": {
    "result_code": 0,
    "refund": {
      "refund_id": 1,
      "amount": "5.00",
      "status": "success",
      "error": 0
    }
  }
}
```

2.6. Проверка статуса возврата

С помощью данного запроса можно опрашивать статус платежа возврата по счету. Для этого необходимо послать GET-запрос по адресу:

https://w.giwi.com/api/v2/prv/{prv_id}/bills/{bill_id}/refund/{refund_id}

- **{prv_id}** - числовой идентификатор провайдера

- **{bill_id}** - уникальный идентификатор счета на стороне провайдера (любая непустая строка до 200 символов).
- **{refund_id}** - идентификатор отмены, уникальный в рамках отмен одного счета.

Параметров запроса нет.

В ответ возвращается результат выполнения операции и все данные об отмене (подробнее см. [ответ сервера](#)).

Пример:

```
GET /api/v2/prv/2042/bills/BILL-1/refund/1
Accept: text/json
Authorization: Basic MjA0MjpwOZxN0Cg==
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

```
HTTP/1.1 200 OK
Content-Type: text/json
```

```
{"response": {
  "result_code": 0,
  "refund": {
    "refund_id": 1,
    "amount": "5.00",
    "status": "success",
    "error": 0
  }
}}
```

2.7. Ответ сервера

Ответ сервера представляет собой объект “response”, состоящий из атрибута “Код результата” и одного из объектов “Счет”, “Возврат по счету”. Результат сериализуется в XML или JSON:

- **Код результата выполнения операции**

Имя параметра	Формат значения	Regexp	Описание
result_code	Натуральное число от 0 до 5000	$\wedge\d{1,4}\$$	Код ошибки при выполнении операции (см. коды ошибок)

Пример сериализации в XML:

```
...
<result_code>0</result_code>
...
```

Пример сериализации в JSON:

```
...
"result_code": 0,
...
```

- **Информация о счете**

Параметры счета:

Имя параметра	Формат значения	Regexp	Описание
bill_id	Любая непустая строка до 200 символов	$\wedge.\{1,200\}\$$	Уникальный идентификатор счета на стороне провайдера
amount	Положительное число, округленное до 2 или 3 знаков после запятой.	$\wedge\d+(\wedge.\d{0,3})?\$$	Сумма возврата. Способ округления зависит от валюты.
ccy	Трёхбуквенная аббревиатура	$\wedge[a-zA-Z]{3}\$$	Идентификатор валюты (Alpha-3 ISO 4217 код)
status	Буквенный идентификатор	$\wedge[a-z]{1,15}\$$	Статус счета (см. статусы счетов)
error	Натуральное число от 0 до 5000	$\wedge\d{1,5}\$$	Код ошибки (см. коды ошибок)
user	Строка вида "tel:{phone_number}" , где {phone_number} – номер телефона в международном формате	$\wedge tel:\wedge\d{1,15}\$$	Идентификатор кошелька пользователя, которому выставляется счет. Представляет собой номер телефона пользователя в международном формате с префиксом "tel:"

comment	Любой текст	$^{\backslash}\{0,255\}\$$	Комментарий к счету
prv_name	Произвольная строка до 100 символов	$^{\backslash}\{1,100\}\$$	Название провайдера, которое отображается пользователю.

Пример сериализации в XML:

```

...
<bill>
<bill_id>bill1234</bill_id>
<amount>99.95</amount>
<ccy>USD</ccy>
<status>paid</status>
<error>0</error>
<user>tel:+79161231212</user>
<comment>Invoice from ShopName</comment>
</bill>
...

```

Пример сериализации в JSON:

```

...
"bill": {
  "bill_id": "bill1234",
  "amount": "99.95",
  "ccy": "USD",
  "status": "paid",
  "error": 0,
  "user": "tel:+79161231212",
  "comment": "Invoice ffrom ShopName"
}
...

```

- **Информация о возврате**

Параметры возврата:

Имя параметра	Формат значения	Regexp	Описание
refund_id	Натуральное число до 4 знаков.	$^{\backslash}d\{1,4\}\$$	Идентификатор отмены, уникальный в рамках отмен одного счета

amount	Положительное число, округленное до 2 или 3 знаков после запятой.	$^{\backslash}d+(\backslash.\backslash d\{0,3\})? \$$	Фактическая сумма возврата
status	Буквенный идентификатор	$^{\backslash}[a-z]\{1,15\} \$$	Статус платежа возврата (см. статусы платежей)
error	Натуральное число от 0 до 5000	$^{\backslash}d\{1,4\} \$$	Код ошибки при проведении платежа
user	Строка вида “tel:{phone_number}”, где {phone_number} – номер телефона в международном формате	$^{\backslash}tel:\backslash+\backslash d\{1,15\} \$$	Идентификатор кошелька пользователя, которому выставляется счет. Представляет собой номер телефона пользователя в международном формате с префиксом “tel:”

Пример сериализации в XML:

```

...
<refund>
<refund_id>12</refund_id>
<amount>99.95</amount>
<status>success</status>
<error>0</error>
<user>tel:+79161231212</user>
</refund>
...

```

Пример сериализации в JSON:

```

...
“refund”: {
  “refund_id”: “12”,
  “amount”: “99.95”,
  “status”: “success”,
  “error”: 0,
  “user”: “tel:+79161231212”
}
...

```

3. Интерфейс провайдера

В протоколе предусмотрена возможность получения провайдером уведомлений о смене статуса выставленного им. Для этого сервис провайдера должен обрабатывать POST-запросы, в которых будет приходить объект “[Счет](#)” с актуальными данными, сериализованный в виде параметров HTTP-запроса. В ответ на запрос должен вернуться код результата обработки уведомления (см. приложения). Ответ должен быть в формате XML. Заголовок Content-Type ответа должен быть «text/xml». В противном случае уведомление будет считаться неуспешным.

Пример HTTP-запроса:

```
POST /qiwi-notify.php HTTP/1.1
Accept: application/xml
Content-type: application/x-www-form-urlencoded
Authorization: Basic MjA0Mjp0ZXN0Cg==

bill_id=BILL-
1&status=paid&error=0&amount=1.00&user=tel%3A%2B79031811737&prv_name=TEST&ccy=RUB&comment=test&command=bill
```

В ответ ожидается следующий XML-документ:

```
HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0"?> <result><result_code>0</result_code></result>
```

Любой код результата обработки уведомления, отличный от 0 (“Успех”), сервер воспринимает как временную ошибку и повторяет запрос с нарастающим интервалом в течение суток (всего 50 попыток) до получения кода 0. Если код 0 так и не был получен, повторные уведомления прекращаются, и на адрес электронной почты провайдера высылается письмо с новым статусом счета и указанием на возможную техническую неисправность в работе сервиса провайдера.

Поскольку при такой схеме уведомлений возможна ситуация, когда на сервер провайдера может прийти два одинаковых уведомления, провайдер должен контролировать, что повторные уведомления не приведут к повторному зачислению денежных средств или оказанию услуги повторно.

В целях ускорения выявления причин временных ошибок рекомендуется возвращать коды результата в соответствии с [таблицей кодов завершения](#).

4. Авторизация при запросах

При регистрации провайдеру выдается дополнительный секретный пароль, используемый при авторизации как при запросах в сторону Visa QIWI Wallet, так и при уведомлениях на сервер провайдера.

4.1. Авторизация при запросах на сервер Visa QIWI Wallet

При авторизации проверяется пароль для доступа к API. Он передается по стандартным правилам basic-авторизации при запросах по HTTP:

К запросу добавляется заголовок “Authorization”, значение которого представляет собой слово “Basic”, пробел и закодированную в BASE64 пару login:password, где login – идентификатор провайдера, password – секретный пароль от API Visa QIWI Wallet.

```
Authorization: Basic bG9naW46cGFzc3dvcmQ=
```

$\text{BASE64}(\text{“login:password”}) = \text{“bG9naW46cGFzc3dvcmQ=”}$

Все запросы в сторону Visa QIWI Wallet шифруются с помощью SSL.

4.2. Авторизация при уведомлениях на сервер провайдера

Предпочтительнее всего, если на сервере провайдера настроено SSL-шифрование. В таком случае передача пароля в любом виде является достаточно безопасной. Если сертификат для SSL-шифрования сгенерирован самостоятельно и не является доверенным со стороны стандартных центров сертификации, этот сертификат можно загрузить в систему Visa QIWI Wallet на партнерском сайте в настройках провайдера. После этого данный сертификат будет считаться доверенным.

Авторизация происходит одним из двух способов:

- Передача пароля в открытом виде

Пара login:password передается по правилам [basic-авторизации](#) и проверяется на сервере провайдера. В случае если при запросе не используется SSL, данный метод не безопасен, поскольку авторизационные данные могут быть перехвачены.

- Использование простой подписи

В HTTP-запрос добавляется заголовок “X-API-Signature”, значение которого представляет собой HMAC-SHA1 хэш-функцию с ключом,

сгенерированным для провайдера на этапе подключения и сообщением, образованным конкатенацией всех параметров запроса в алфавитном порядке с использованием разделителя “|”:

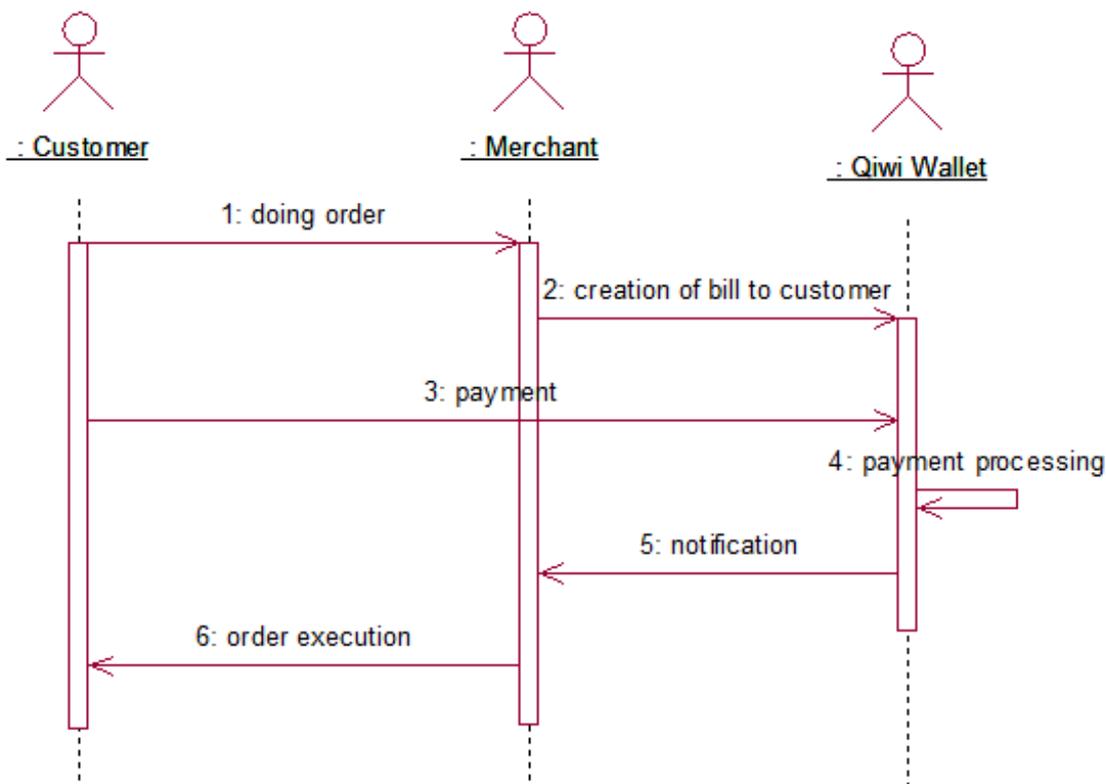
```
sign = HMAC-SHA1(key, {amount}|{bill_id}|{ccy}|{command}|{comment}|{error}|{prv_name}|{status}|{user})
```

- **key** – секретный пароль от API Visa QIWI Wallet
- **{parameter}** – значение соответствующего параметра счета из уведомления.
- **sign** – base64-кодированный результат функции

Все аргументы функции HMAC-SHA1 преобразуются из строки в байты с использованием кодировки UTF-8. Байтовый результат выполнения функции HMAC-SHA1 кодируется в Base-64. Сумма должна быть отформатирована с двумя знаками после запятой (например, “10.00”).

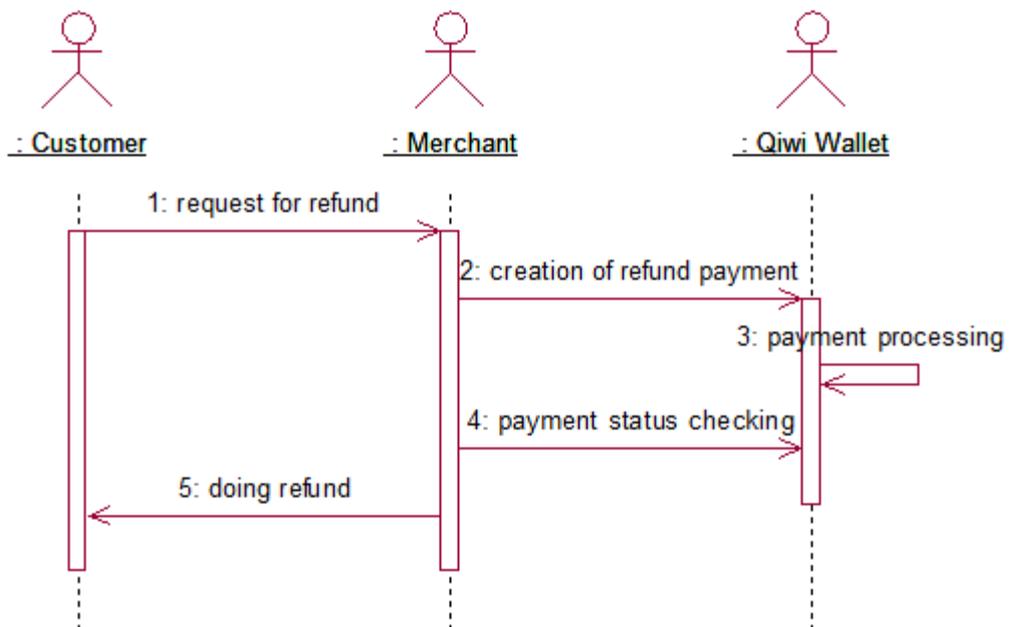
5. Стандартная схема работы

Пользователь формирует заказ на сайте провайдера, далее провайдер посылает [запрос на создание счета](#) на сервер Visa QIWI Wallet. В случае успеха пользователь должен авторизоваться в системе Visa QIWI Wallet через любой из интерфейсов и оплатить счет. После проведения платежа об оплате счета система Visa QIWI Wallet высылает [уведомление](#) на сервер провайдера об оплате данного счета, либо, если пользователь отклонил счет, о неоплате. Далее, если счет был оплачен, провайдер исполняет заказ пользователя.



В любой момент сервер провайдера может [запросить статус созданного счета](#), либо [отменить счет](#) (при условии, что он еще не был оплачен).

При необходимости возврата пользователю всей суммы оплаченного счета или её части провайдер должен послать на сервер Visa QIWI Wallet [запрос на осуществление возврата](#). Далее стоит убедиться, что платёж проведен успешно, для этого можно периодически опрашивать сервис Visa QIWI Wallet о текущем статусе платежа возврата до получения финального статуса.



Данный сценарий можно повторять несколько раз до тех пор, пока счет не будет полностью отменен (возвращена вся сумма).

6. Приложения

6.1. Статусы счетов

Код статуса	Описание	Финальный статус
waiting	Счет выставлен, ожидает оплаты.	Нет
paid	Счет оплачен.	Да
rejected	Счет отклонен.	Да
unpaid	Ошибка при проведении оплаты. Счет не оплачен.	Да
expired	Время жизни счета истекло. Счет не оплачен.	Да

6.2. Статусы платежей

Код статуса	Описание	Финальный статус
processing	Платеж в проведении	Нет
success	Платеж проведен	Да
fail	Платеж неуспешен	Да

6.3. Коды ошибок

Код ошибки	Описание	Фатальность*
0	Успех	Нет
5	Неверный формат параметров запроса	Да
13	Сервер занят, повторите запрос позже	Нет
150	Ошибка авторизации	
210	Счет не найден	Да
215	Счет с таким bill_id уже существует	Да
241	Сумма слишком мала	Да
242	Сумма слишком велика	Да
298	Кошелек с таким номером не зарегистрирован	Да
300	Техническая ошибка	Нет

* фатальность – при повторном запросе результат не изменится (ошибка не временная)

6.4. Коды завершения уведомлений

Код завершения	Описание
0	Успех
5	Ошибка формата параметров запроса
13	Ошибка соединения с базой данных
150	Ошибка проверки пароля
151	Ошибка проверки подписи
300	Ошибка связи с сервером