



VISA QIWI WALLET PULL PAYMENTS API

ver. 2.1

USER GUIDE
ver. 2.0

MOSCOW
8-495-783-5959

RUSSIA
8-800-200-0059

FAX
8-495-926-4619

WEB
WWW.QIWI.COM

CONTENT

1.	INTRODUCTION	3
1.1.	PURPOSE OF THE API.....	3
1.2.	THINGS TO KNOW ABOUT VISA QIWI WALLET	3
2.	TYPICAL INTERACTION FLOW	4
2.1.	INVOICE ISSUE	4
2.2.	REFUND SCENARIO	4
3.	AUTHORIZATION	6
3.1.	AUTHORIZATION WHEN SENDING REQUESTS TO VISA QIWI WALLET SERVER	6
3.2.	AUTHORIZATION WHEN RECEIVING NOTIFICATIONS ON THE MERCHANT'S SERVER.....	6
4.	VISA QIWI WALLET INTERFACE	8
4.1.	INVOICE WEB FORM	8
4.2.	CREATING AN INVOICE.....	9
4.3.	REQUESTING INVOICE STATUS	10
4.4.	REDIRECTION FOR INVOICE PAYMENT.....	11
4.5.	CANCELLING UNPAID INVOICES	12
4.6.	REFUNDS.....	13
4.7.	REFUND STATUS VERIFICATION	14
4.8.	SERVER RESPONSE	15
5.	REQUIREMENTS TO MERCHANT INTERFACE.....	17
6.	APPENDICES	18
6.1.	INVOICE STATUS	18
6.2.	PAYMENT STATUS.....	18
6.3.	ERROR CODES.....	18
6.4.	NOTIFICATION CODES	19
6.5.	EXAMPLE OF WEB-INTERFACE FOR INVOICE.....	19

1. INTRODUCTION

1.1. Purpose of the API

This API is to be implemented by a merchant/PSP (hereinafter as "merchant") to support Visa QIWI Wallet Pull Payments. Pull payments are those initiated from the merchant's website, in contrast to Push payments that are initiated in Visa QIWI Wallet interfaces such as web, mobile, terminal, etc.

1.2. Things to Know About Visa QIWI Wallet

Visa QIWI Wallet is an online and mobile payment service in Russia and other countries. Visa QIWI Wallet is available online, with mobile applications, and with QIWI payment kiosks.

Visa QIWI Wallet uses mobile phone number as user ID. Passwords are sent by SMS.

Users can fund their payments with Visa QIWI Wallet from the prepaid balance of Visa QIWI Wallet account, from the mobile phone account prepaid balance, with Visa/MasterCard debit or credit card, or in cash with a QIWI kiosk.

Visa QIWI Wallet server interacts with the merchant's server over HTTP protocol.

Requests from the merchants to QIWI are sent in the format of the HTTP-request parameters encoded by UTF-8. In response, the data is returned in one of two formats in accordance with the value of the "Accept" header, which is transmitted in the request:

- XML (value of the "Accept" header: "application/xml", "text/xml");
- JSON (value of the "Accept" header: "application/json", "text/json").

For notifications to the merchant's server the data is transmitted as "application/x-www-form-urlencoded" content type encoded by UTF-8. Response must be in XML format.

To receive notifications merchant must whitelist following IP subnets connected by 80, 443 ports exclusively:

- 91.232.230.0/23
- 79.142.16.0/20

For security purposes, all data transmitted to QIWI Servers is encrypted using SSL. Unencrypted HTTP requests are not supported. Authorization on Visa QIWI Wallet side is performed by login and password for API access (for details see chapter "[Authorization](#)").

SSL might also be used for encryption of notifications to the merchant's servers (it is possible to use self-generated certificates). Otherwise, it is possible to use simplified signature algorithm based on HMAC-SHA1 (for details see chapter "[Authorization](#)").

Merchant must check validity of Visa QIWI Wallet's server certificate using standard algorithm for certificate validation.

2. TYPICAL INTERACTION FLOW

2.1. Invoice issue

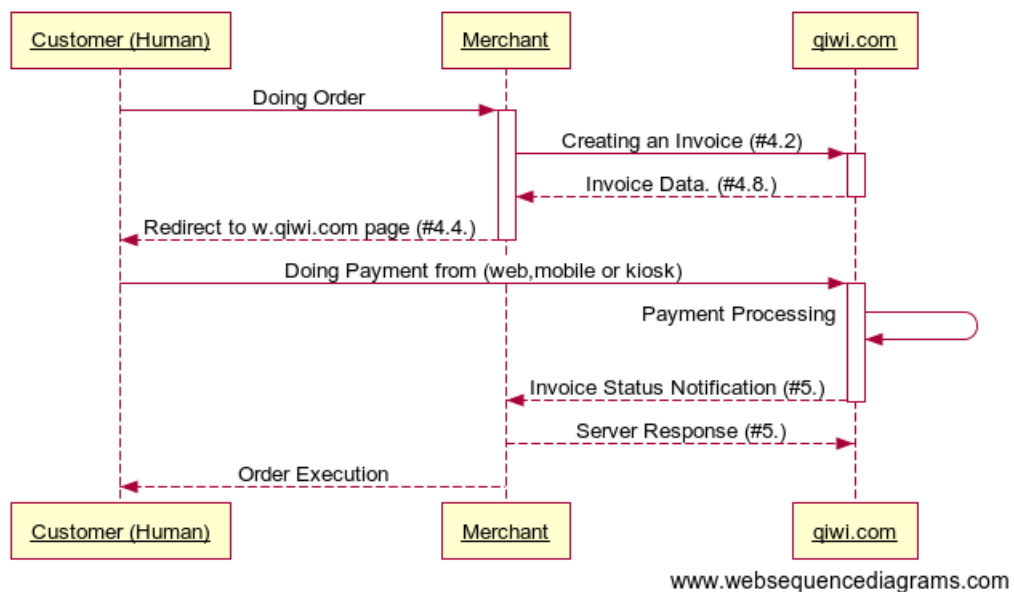
User submits an order on the merchant's website. Then the following invoice scenarios are possible:

1. Merchant [creates invoice form](#) on Visa QIWI Wallet site by a HTTP GET-request. It is the simplest way to make an invoice. No authorization is required in this case.
2. Merchant [sends invoice request](#) to Visa QIWI Wallet server. This request requires [authorization](#).
Merchant is recommended to [redirect to checkout page](#) on Visa QIWI Wallet site when request is completed.

Once the invoice is created, user should log on to Visa QIWI Wallet and pay the invoice. Once invoice is paid, QIWI sends to the merchant's server a [notification](#) on the invoice status: either paid or cancelled by the user. Authorization on the merchant's side is required for notifications.

Finally, merchant delivers ordered services/goods when the invoice gets paid.

Fig. 1. Invoice issuing scenario diagram



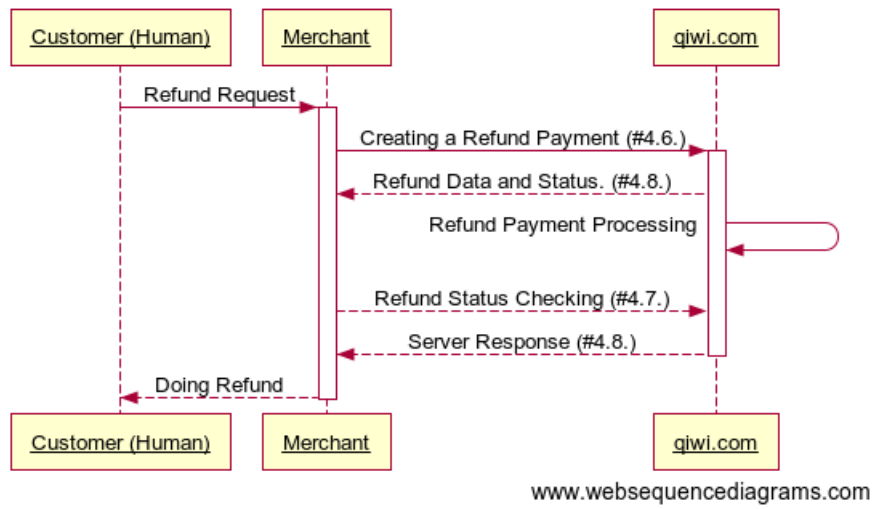
Merchant can [request current status of the created invoice](#), or [cancel invoice](#) (provided that it has not been paid yet) at any moment.

2.2. Refund scenario

If it is necessary to refund a part of the invoice amount or the full amount, merchant has to send a [request for refund](#) to Visa QIWI Wallet server.

To make sure that the payment refund has been successfully processed merchant can periodically request the invoice [refund status](#) until the final status is received.

Fig. 2. Refund scenario diagram



This scenario can be repeated multiple times until the invoice is completely refunded (whole invoice amount has been returned to the user).

Refund requests to Visa QIWI Wallet server require [authorization](#).

3. AUTHORIZATION

3.1. Authorization when sending requests to Visa QIWI Wallet server

Upon registration the merchant will receive an additional password which should be used for authorization when sending requests to Visa QIWI Wallet. When generated on ishop.qiwi.com site, the password is linked to specific identifier for Visa QIWI Wallet API (API ID).

The merchant can fully control authorization data by generating new password and API ID couples as well as removing old ones. Password and ID are generated in **Merchants details** → **Authentication details for all protocols** section of ishop.qiwi.com web-site by clicking the [Generate new ID](#) link.

The password to the API is verified during the authorization. The password is transmitted using the standard rules of basic-authorization for HTTP-requests. The header "Authorization" is added to the request. The value of this parameter is composed of the word "Basic", blank character and encrypted BASE64 pair `login:password`, where login – merchant's unique identifier in Visa QIWI Wallet API (API ID), password – password to Visa QIWI Wallet API, corresponding to this API ID:

```
Authorization: Basic bG9naW46cGFzc3dvcmQ=
```

Where:

```
BASE64("login:password") = "bG9naW46cGFzc3dvcmQ="
```

The transmission of the password is secured as all requests to QIWI are SSL-encrypted.

Example of using authorization data is given in [PHP script for invoice issuing](#).

3.2. Authorization when receiving notifications on the merchant's server

It is preferred that the merchant's server accepts notifications from Visa QIWI Wallet with HTTPS protocol. In this case the password transmission is secured enough. Merchant's server should support SSL-encryption and client SSL certificate verification.

If the SSL-certificate is self-generated and is not issued by one of the standard certification centers, this certificate can be uploaded to the Visa QIWI Wallet server via the merchant's console (**Certificate** field in **Merchant details** → **Pull protocol settings** section of ishop.qiwi.com web-site). Certificate must be in one of the following formats:

- PEM (text file with .pem extension) – (Privacy-enhanced Electronic Mail) Base-64 encoded DER certificate placed between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----" strings.
- DER (binary file with .cer, .crt, .der extensions) – usually in binary DER format, though PEM certificates are also accepted with this extensions.

Then the merchant's certificate becomes trusted after the upload.

Merchant's server should perform authorization of the notifications by "login:password" secret pair:

- Login – merchant's Shop ID, as displayed in **Merchant details** → **HTTP-protocol settings** section of ishop.qiwi.com web-site.
- Password is issued on merchant's registration. Merchant should change it immediately in **Merchant details** → **Pull protocol settings** section of ishop.qiwi.com web-site (**Change password** button). If necessary, merchant may reset password for notifications at any time.

Merchant's server should perform authorization of the notifications by one of the following ways:

1. Plain text password transmission. No additional settings are necessary.

The pair "login:password" is transmitted according to the rules of [basic-authorization](#) and verified on the merchant's server. If the request is not SSL-encrypted, the transmission is not secured and the data becomes vulnerable to interception.

2. Simplified signature algorithm. Merchant should enable **Sign** flag in **Merchant details** → **Pull protocol settings** section of ishop.qiwi.com web-site.

The header "X-Api-Signature" is added to the HTTP-request. The value of the header is composed of the HMAC-SHA1 function using password for notifications (see above) as a key and all invoice parameters placed in alphabetical order and separated by "|" as message. For example:

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

Where:

- **key** – password used to access Visa QIWI Wallet API;
- **{parameter}** – value of the respective parameter of the invoice taken from the notification;
- **sign** – base64 encoded result string.

All parameter values included into the sign should be treated as strings. All arguments of HMAC-SHA1 function are converted to bytes using UTF-8 character encoding. Byte result of HMAC-SHA1 function is encoded in Base-64 string.

WARNING



As new parameters of the invoice might be introduced on Visa QIWI Wallet side, a list of parameters in HTTP-request should not be fixed on the merchant's side and should be taken from the request itself.

4. VISA QIWI WALLET INTERFACE

4.1. Invoice Web Form

Making invoice in Visa QIWI Wallet site is performed by HTTPS protocol via GET request with a set of [parameters](#).

Server URL address: <https://w.qiwi.com/order/external/create.action>

In GET request you should send the following bill parameters:

Parameter	Description	Format
txn_id	Unique invoice number in online store. It is used to identify specific invoice.	A string of Latin letters and digits (without space). Max length is 30 bytes.
from	Merchant identifier (Shop ID). Required parameter. Identifier is specified in HTTP protocol setting section of merchant's cabinet ishop.qiwi.com .	Integer positive. Required.
to	Visa QIWI Wallet client phone number to make invoice If absent, web-form with phone number input field would be displayed to client	String of phone number by international standard. «+» symbol can be omitted or specified explicitly as well as encoded "%2B" Required.
summ	Amount of invoice. Required parameter.	Digit with point as a separator. Required.
currency	Invoice currency. Required parameter. Any currency may be used if specified in agreement with QIWI.	String or number of currency code by ISO 4217 Required.
successUrl	Client is redirected to the specified URL when invoice is successfully paid. May be parameter or anchor	URL by HTTP specification
failUrl	Client is redirected to the specified URL when invoice payment is unsuccessful. May be parameter or anchor	URL by HTTP specification
lifetime	Invoice lifetime (in minutes), counting from invoice creation time. When time is expired, invoice cannot be paid and it would be cancelled.	Integer positive
comm	Merchant commentary to invoice	String with no more than 255 symbols
iframe	This parameter means that invoice page would be opened in "iframe". Page view is tight and easily framed into provider site	true/false "false" by default
target	This parameter means that hyperlink specified in <i>successUrl</i> / <i>failUrl</i> parameter opens in "iframe"	"iframe" string or empty

In response, server displays invoice web page (or page with error result) on Visa QIWI Wallet website. Page contains invoice number automatically assigned in Visa QIWI Wallet system.

Examples of requests to create invoice web-form:

Example 1. HTTP requests for invoicing

```
https://w.qiwi.com/order/external/create.action?comm=test&from=000000&summ=1.01&currency=RUB&to=71234567890

https://w.qiwi.com/order/external/create.action?comm=test&from=000000&summ=1.01&currency=USD&to=+71234567890

https://w.qiwi.com/order/external/create.action?comm=test&txn_id=0000&from=000000&summ=1.11&successUrl=http%3A%2F%2Ftest.ru%3Fcurrency=643&to=%2B71234567890
```

4.2. Creating an Invoice

To create an invoice to the user's Visa QIWI Wallet implicitly (without loading web form) the merchant has to send PUT-request to the following URL:

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

where:

- **{prv_id}** – merchant's unique ID (numeric value);
- **{bill_id}** - unique invoice identifier generated by the merchant (any non-empty string of up to 200 characters).

Request parameters:

Name	Value format	Regexp	Description
user	String of the form "tel:phone_number", where "phone_number" - number in international format	^tel:\+\d{1,15}\$	The QW user's ID, to whom the invoice is issued. It is the user's phone number with "tel:" prefix. Required.
amount	A positive number rounded up to 2 or 3 decimal places after the comma	^\d+(\.\d{0,3})?\$	The invoice amount. The rounding up method depends on the invoice currency. Required.
ccy	Three-letter abbreviation	^[a-zA-Z]{3}\$	Currency identifier (Alpha-3 ISO 4217 code). Required.
comment	Any text	^\.{0,255}\$	Comment to the invoice. Required.
lifetime	Date, up to the seconds, in ISO 8601 format	^\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}\$	Date and time up to which the invoice is available for payment. If the invoice is not paid by this date it will become void and will be assigned a final status. Required.
pay_source	"mobile", "qw"	^((mobile) (qw)){1}\$	If the value is "mobile" the user's MNO balance will be used as a funding source. If the value is "qw", any other funding source is used available in QW interface. If parameter isn't present, value "qw" is assumed

Name	Value format	Regexp	Description
prv_name	Any text, not more than 100 symbols		Merchant's name which will be shown to the user.

The response will contain the result code and in case if the invoice was successfully created – all information about the invoice (for more details see [server response](#)).

Sample request (JSON response):

```
PUT /api/v2/prv/2042/bills/BILL-1
Accept: text/json
Authorization: Basic MjA0MjpwOjZxN0Cg==
Content-Type: application/x-www-form-urlencoded; charset=utf-8

user=tel%3A%2B79031234567%26amount=10.0%26ccy=RUB%26
comment=Order #1234 at hosting.com%26lifetime=2012-11-25T09%3A00%3A00
```

Response:

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

{"response": {
  "result_code": 0,
  "bill": {
    "bill_id": "BILL-1",
    "amount": "10.00",
    "ccy": "RUB",
    "status": "waiting",
    "error": 0,
    "user": "tel:+79031234567",
    "comment": "Order #1234 at hosting.com"
  }
}}
```

4.3. Requesting Invoice Status

Merchant can request the current status of the invoice by sending GET-request to the following URL: https://w.qiwi.com/api/v2/prv/{prv_id}/bills/{bill_id}

where:

- **{prv_id}** - merchant's unique ID (numeric value);
- **{bill_id}** - unique invoice identifier generated by the merchant (non-empty string of up to 200 characters).

There are no parameters for the request.

The response will contain the result code of the operation, and in case of successful execution – all information about the invoice (for more details see [server response](#)).

Sample request (JSON response):

```
GET /api/v2/prv/2042/bills/BILL-1
Accept: text/json
Authorization: Basic MjA0MjpwOjZxN0Cg==
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

Response:

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

{"response": {
  "result_code": 0,
  "bill": {
    "bill_id": "BILL-1",
    "amount": "10.00",
    "ccy": "RUB",
    "status": "waiting",
    "error": 0,
    "user": "tel:+79031234567",
    "comment": "Order #1234 at hosting.com"
  }
}}
```

4.4. Redirection for Invoice Payment

Merchant may offer a Visa QIWI Wallet user to pay the invoice immediately by redirecting to the Visa QIWI Wallet payment page or by opening it in "iframe" to one of the following URLs:

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

The two URLs work exactly in the same way except that the second one appears more compact and can be embedded conveniently within the merchant's site.

Both URLs accept the same set of parameters:

Name	Type	Description	Example
shop	string	Merchant's ID in Visa QIWI Wallet system, corresponds to {prv_id} parameter used to create the bill	123
transaction	string	Invoice ID generated by the merchant, corresponds to {bill_id} parameter used to create the bill	abcde12345
successUrl	URL-encoded string	The URL to which the payer will be redirected in case of successful payment. URL must be within merchant's site	http%3A%2F%2Fmystore.com%2Fsuccess%3Fa%3D1%26b%3D2
failUrl	URL-encoded string	The URL to which the payer will be redirected in case of unsuccessful payment URL must be within merchant's site	http%3A%2F%2Fmystore.com%2Ffail%3Fa%3D1%26b%3D2

Optionally, Visa QIWI Wallet will redirect back to URL specified in **successUrl** or **failUrl** (if any) with an additional parameter. The name of the parameter is **order** and the value is the invoice ID taken from the **transaction** parameter of the initial request. Using this parameter, merchant can render the final page depending on the order details.

Example:

- Merchant redirects to URL:

<https://w.qiwi.com/order/external/main.action?shop=2042&transaction=123123123&successUrl=http://mystore.com/success?a=1&b=2&failUrl=http://mystore.com/fail?a=1&b=2>.

- User pays the invoice using one of the available payment options.
- Visa QIWI Wallet redirects user to <http://mystore.com/success?a=1&b=2&order=123123123>.

WARNING



Don't rely on redirection to **successUrl** as an evidence of successful payment, because it can be simulated at the user's end. It serves only as a convenient interface solution. Wait for the Visa QIWI Wallet's [notification](#) to the merchant's site before complete the user order.

4.5. Cancelling Unpaid Invoices

Merchant can cancel a previously issued invoice by sending a PATCH-request to the following URL (provided that the invoice has not been paid):

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

where:

- **{prv_id}** - merchant's respective unique ID (numeric value);
- **{bill_id}** - unique invoice identifier generated by the merchant (non-empty string of up to 200 characters).

Request parameters:

Name	Value format	Regexp	Description
status	"rejected"	^rejected\$	New invoice status (cancelled). Required.

The response will contain the result code of the operation, and in case of successful execution – all information about the invoice (for more details see [server response](#)).

Sample request (JSON response):

```

PATCH /api/v2/prv/2042/bills/BILL-2
Accept: text/json
Authorization: Basic MjA0MjBpOZXXN0Cg==
Content-Type: application/x-www-form-urlencoded; charset=utf-8

status=rejected
    
```

Response:

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

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

4.6. Refunds

Merchant can process a full or partial refund to user's Visa QIWI Wallet account using the PUT-request described below. This request creates a reversed transaction for the initial one.

Merchant can create several refund operations for the same initial invoice provided that:

- Amount of all refund operations does not exceed initial invoice amount.
- Different refund IDs are used for different refund operations of the same invoice (see below).

In case when the transmitted amount exceeds the initial invoice amount or the amount left after the previous refunds, only the remaining amount will be refunded.

To process the request, merchant has to send PUT-request to the following URL:

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

where:

- **{prv_id}** - merchant's unique ID (numeric value);
- **{bill_id}** - unique invoice identifier generated by the merchant (non-empty string of up to 200 characters);
- **{refund_id}** – refund identifier, a number specific to a series of refunds for the invoice **{bill_id}** (random integer up to 9 digits).

Request parameters:

Name	Value format	Regexp	Description
amount	A positive number rounded up to 2 or 3 decimal places after the comma	$^{\wedge}\d+(\.\d{0,3})?\$$	The refund amount should be less or equal to the amount of the initial transaction. The rounding up method depends on the invoice currency. Required.

The response will contain the result code of the operation, and in case of successful execution – all information about the refund of the invoice (for more details see [server response](#)).

Sample request (JSON response):

```
PUT /api/v2/prv/2042/bills/BILL-1/refund/1
Accept: text/json
Authorization: Basic MjA0MjpwOZlXN0Cg==
Content-Type: application/x-www-form-urlencoded; charset=utf-8

amount=5.0
```

Response:

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

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

4.7. Refund Status Verification

Merchant can verify status of the refund by sending GET-request to the following URL:

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

Where:

- **{prv_id}** - merchant's unique ID (numeric value);
- **{bill_id}** - unique invoice identifier generated by the merchant (non-empty string of up to 200 characters);
- **{refund_id}** – refund identifier, unique number from a series of refunds processed for the invoice **{bill_id}** (see section 4.6).

There are no parameters for the request.

The response will contain the result code of the operation, and in case of successful execution – all information about the refund of the invoice (for more details see [server response](#)).

Sample request (JSON response):

```
GET /api/v2/prv/2042/bills/BILL-1/refund/1
Accept: text/json
Authorization: Basic MjA0MjpwOZlXN0Cg==
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

Response:

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

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

4.8. Server Response

The server response represents an object "Response", which is composed of "Result code" element and one of the following objects depending on the request type: "Invoice", "Refund". The result is serialized in XML or JSON:

- The result code of an operation:

Name	Value format	Regexp	Description
result_code	Integer from 0 to 5000	^\d{1,4}\$	Error code received after the operation execution (see error codes)

Example of serialization in XML:

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

Example of serialization in JSON:

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

- Information on invoice.

Parameters of the invoice:

Name	Value format	Regexp	Description
bill_id	Any non-empty string up to 200 characters	^\.{1,200}\$	Unique invoice identifier generated by the merchant
amount	A positive number rounded up to 2 or 3 decimal places after the comma	^\d+(\.\d{0,3})?\$/	The invoice amount. The rounding up method depends on the invoice currency
ccy	Three-letter abbreviation	^[a-zA-Z]{3}\$	Currency identifier (Alpha-3 ISO 4217 code)
status	Alphanumeric identifier	^[a-z]{1,15}\$	Invoice status (see Invoice status .)
error	Integer from 0 to 5000	^\d{1,5}\$	Error code (see error codes)
user	String of the form "tel:phone_number", where phone_number – phone number in international format	^tel:\+\d{1,15}\$	The QW user's ID, to whom the invoice is issued. It is the user's phone number with "tel:" prefix.
comment	Any text	^\.{0,255}\$	Comment to the invoice

Example of XML serialization:

```
...
<bill>
<bill_id>bill11234</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>
...
```

Example of JSON serialization:

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

- Information on refund.

Parameters of the refund:

Name	Value format	Regexp	Description
refund_id	Integer up to 9 numbers	^\d{1,9}\$	The refund identifier, unique number in a series of refunds processed for a particular invoice.
amount	A positive number rounded up to 2 or 3 decimal places after the comma	^\d+(\.\d{0,3})?\$	The actual amount of the refund
status	Alphanumeric identifier	^[a-z]{1,15}\$	Refund status (see Payment statuses)
error	Integer from 0 to 5000	^\d{1,4}\$	Error code (see error codes)
user	String of the form "tel:phone_number", where phone_number – phone number in international format	^tel:\+\d{1,15}\$	The QW user's ID, to whom the invoice is issued. It is the user's phone number with "tel:" prefix

Example of XML serialization:

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

Example of JSON serialization:

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


5. REQUIREMENTS TO MERCHANT INTERFACE

This API enables merchant to receive the notifications on invoice status change. To receive these notifications merchant's server should be able to accept specific HTTP-requests on ports 80, 443. Notifications are POST-requests with object "[Invoice](#)" containing all relevant data, serialized in the form of HTTP-request parameters plus parameter "command", which always has "bill" value (application/x-www-form-urlencoded).

Requests must be authorized in accordance with requirements specified in section [Authorization when receiving notifications on the merchant's server](#).

In response to the request the result code will be returned (see [Notification codes](#)). The response should be in XML. "Content-type" header must be "text/xml" otherwise notification is treated as unsuccessful on Visa QIWI Wallet side.

Example request to merchant's server:

```
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
```

The following response is expected (result code may vary):

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

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

Result code must be placed in `result_code` tag. Any notification result code other than 0 ("Success"), will be treated by Visa QIWI Wallet server as a temporary error. Thus the server will continue repeating requests with increased time intervals within next 24 hours (50 attempts in total) until it gets code 0 ("Success"). If the code 0 ("Success") has not been received within 24 hours, Visa QIWI Wallet server will stop sending the requests and will send an email to the merchant with new [Invoice status code](#) and indication on the possible technical issues on the merchant's server side.

Because there could be several notification attempts for one invoice, merchant must not deposit money to user or provide service twice.

In order to help in identifying the reasons of notification errors we recommend that the result codes returned by the merchant are in accordance with [Notification codes table](#).

6. APPENDICES

6.1. Invoice status

Status code	Description	Final status?
waiting	Invoice issued, pending payment	No
paid	Invoice has been paid.	Yes
rejected	Invoice has been rejected.	Yes
unpaid	Payment processing error. Invoice has not been paid.	Yes
expired	Invoice expired. Invoice has not been paid.	Yes

6.2. Payment status

Status code	Description	Final status?
processing	Payment is pending	No
success	Payment is successful	Yes
fail	Payment is unsuccessful	Yes

6.3. Error codes

Error code	Description	Fatal? *
0	Success	
5	Incorrect data in the request parameters	Yes
13	Server is busy, try again later	No
78	Operation is forbidden	Yes
150	Authorization error (e.g. invalid login/password)	Yes
152	Protocol is not enabled or protocol is disabled	No
210	Invoice not found	Yes
215	Invoice with this bill_id already exists	Yes
241	Invoice amount less than minimum	Yes
242	Invoice amount greater than maximum (15 000 RUB)	Yes
298	User not registered	Yes
300	Technical error	No

Error code	Description	Fatal? *
303	Wrong phone number	Yes
316	Authorization by provider which is blocked	No
319	No rights for the operation	No
341	Required parameter is incorrectly specified or absent in the request	Yes
1001	That currency is not allowed for the provider	Yes
1003	No convert rate for these currencies	No
1019	Unable to determine wireless operator for mobile commerce	Yes

* Fatal means the result will not change with the second and subsequent requests (error is not temporary)

6.4. Notification codes

Result code	Description
0	Success
5	The format of the request parameters is incorrect
13	Database connection error
150	Incorrect password
151	Signature authorization failed
300	Server connection error

6.5. Example of web-interface for invoice

This example written on PHP is given to demonstrate using provider's authorization parameters, i.e. shop ID, API ID and password for the API ID.

```
<?php
//Shop identifier from Merchant details page
//https://ishop.qiwi.com/options/merchants.action
$SHOP_ID = "21379721";
//API ID from Merchant details page
//https://ishop.qiwi.com/options/merchants.action
$REST_ID = "62573819";
//API password from Merchant details page
//https://ishop.qiwi.com/options/merchants.action
$PWD = "*****";
//Invoice ID
$BILL_ID = "99111-ABCD-1-2-1";
$PHONE = "79191234567";

$data = array(
    "user" => "tel:+" . $PHONE,
    "amount" => "1000.00",
    "ccy" => "RUB",
```

```

    "comment" => "Good choice",
    "lifetime" => "2015-01-30T15:35:00",
    "pay_source" => "qw",
    "prv_name" => "Special packages"
);

$ch = curl_init('https://w.qiwi.com/api/v2/prv/'. $SHOP_ID.'/bills/'. $BILL_ID);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'PUT');
curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($data));
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_USERPWD, $REST_ID.':'.$PWD);
curl_setopt($ch, CURLOPT_HTTPHEADER,array (
    "Accept: application/json"
));
$results = curl_exec ($ch) or die(curl_error($ch));
echo $results;
echo curl_error($ch);
curl_close ($ch);
//Необязательный редирект пользователя
$url = 'https://w.qiwi.com/order/external/main.action?shop='.$SHOP_ID.'&
transaction='.$BILL_ID.'&successUrl=http%3A%2F%2Fieast.ru%2Findex.php%3Froute%3D
payment%2Fqiwi%2Fsuccess&failUrl=http%3A%2F%2Fieast.ru%2Findex.php%3Froute%3D
payment%2Fqiwi%2Ffail&qiwi_phone='.$PHONE;
echo '<br><br><b><a href="'. $url.'">Redirect link to pay for invoice</a></b>';
?>

```