Documentation on Alfa-Bank's Faster Payments System

only for one-phase payments

**Table of contents**

[1 Terms and abbreviations 3](#_Toc127951219)

[2 Payment via FPS 4](#_Toc127951220)

[2.1 One-phase payment on the payment page 4](#_Toc127951221)

[2.2 One-phase payment with data entry on the merchant's side 5](#_Toc127951222)

[2.3 Payment via the "instant invoice" service 6](#_Toc127951223)

[2.4 Description of payment API 6](#_Toc127951224)

[2.4.1 SOAP 6](#_Toc127951225)

[2.4.2 REST 13](#_Toc127951226)

[3 Refund 20](#_Toc127951227)

[3.1 Refund via API 20](#_Toc127951228)

[3.3.1 Scenario of a refund via API 20](#_Toc127951229)

[3.1.2 Description of the API for refund 21](#_Toc127951230)

[3.2 Refund via PA 23](#_Toc127951231)

[4 Viewing in PA of orders paid via the FPS 24](#_Toc127951232)

[4.1 Search for orders 24](#_Toc127951233)

[4.2 Order history 24](#_Toc127951234)

[5 Fiscalization of transactions 25](#_Toc127951235)

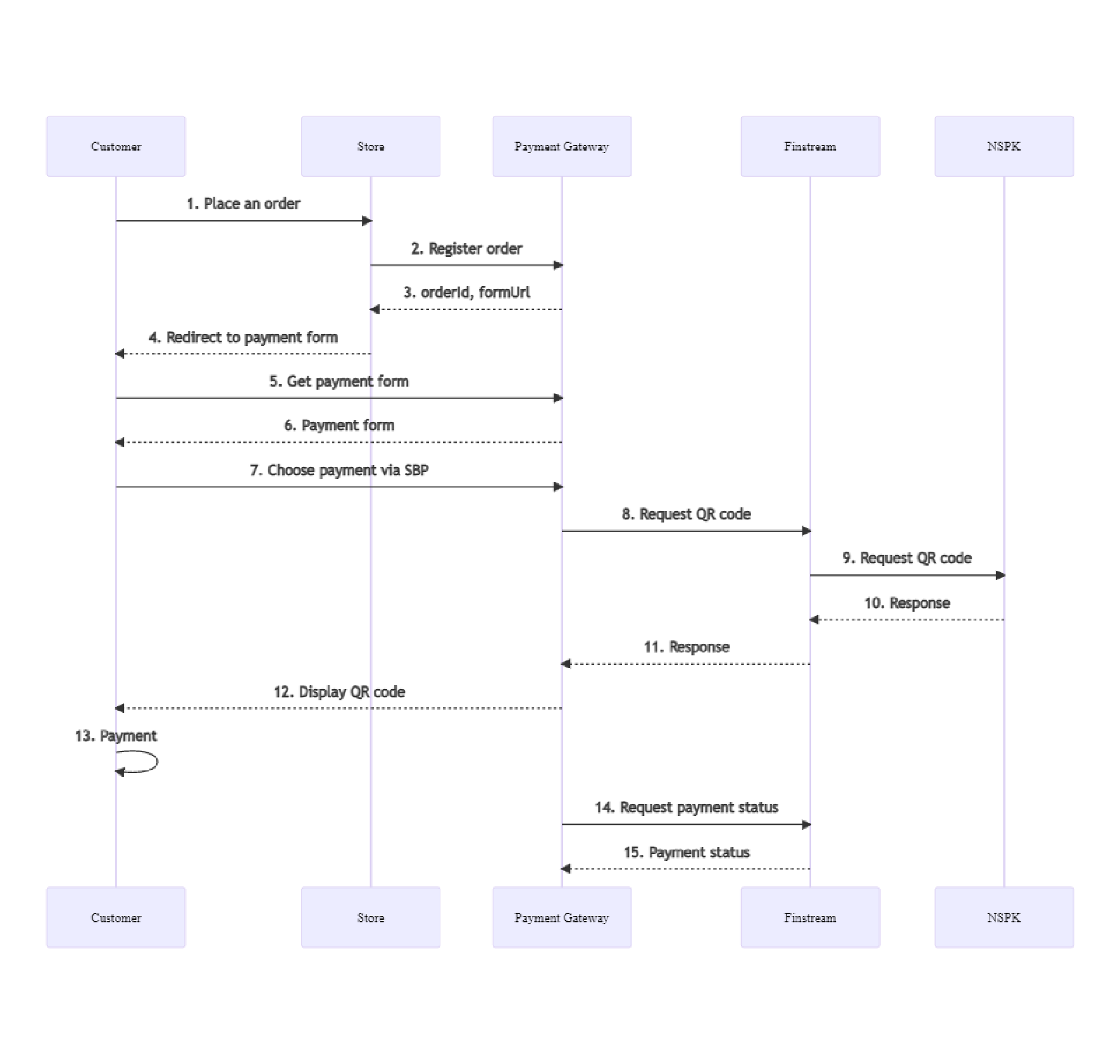
[6 Stubbing mode on the testing fixture 26](#_Toc127951236)

1. Terms and abbreviations

|  |  |
| --- | --- |
| **Term** | **Description** |
| Bank | An acquiring bank that implements and operates the Payment Gateway. |
| Merchant, Store | A retail and service outlet (RSO) that sells goods or provides services through a website. |
| FPS | The Faster Payments System of the National Payment Card System (NPCS) of the Central Bank of the Russian Federation. |
| One-phase payment | An operation to pay for goods/services made via the Internet using bank cards, which does not require additional confirmation |

# 

1. Payment via FPS
   1. One-phase payment on the payment page



Description:

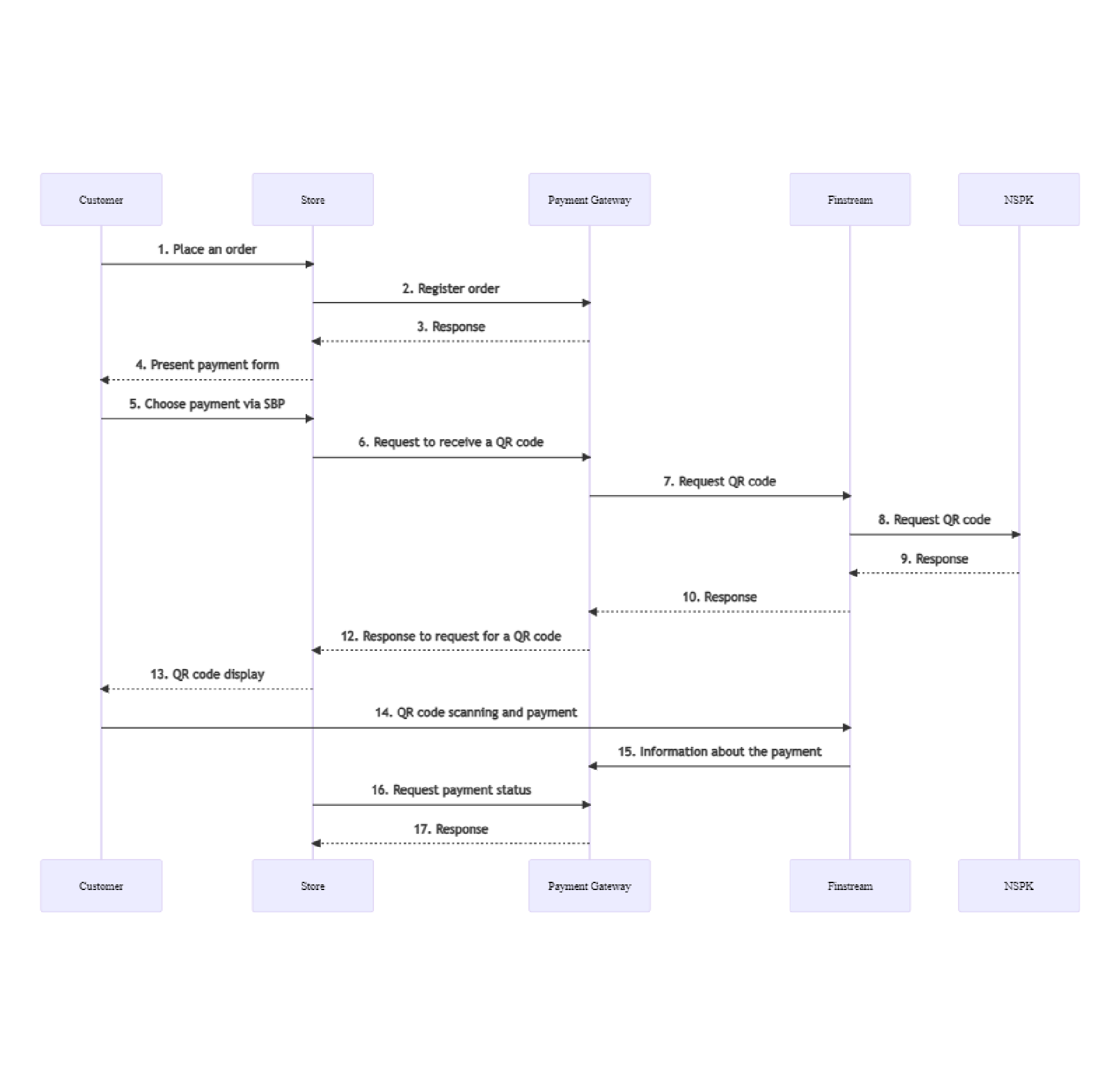
* + 1. The buyer places an order and proceeds to payment.
    2. The Merchant sends a request for order registration to the Payment gateway. The specification of requests is presented in the following sections:
       - "[2.4.1.1. Request for order registration (SOAP](#_bookmark2))";
       - "[2.4.2.1. Request for order registration (REST](#_bookmark5))".
    3. In response to the registration request, the Payment gateway returns a unique order identifier in the payment system (in the orderId parameter) and the URL to which the Client must be redirected to receive the payment form (in the formUrl parameter).
    4. The Merchant's system must redirect the Client's browser to the URL received from the Payment gateway in the formUrl parameter in response to the order registration request.
    5. The Client's browser requests the page at the specified URL.
    6. The browser displays a payment page where the Client chooses to pay via the FPS.
    7. The Payment page requests a QR code in the Payment gateway.

8-11. The payment gateway sends the QR code request to the FPS system.

1. The QR code is displayed to the client on the payment page.
2. The client reads the QR code using specialized software and makes the payment.

14-15. The payment gateway polls the FPS system for the payment status and, if the status is successful, completes the payment.

* 1. One-phase payment with data entry on the merchant's side



Description:

* + 1. The buyer places an order and proceeds to payment.
    2. The Merchant sends a request for order registration to the Payment gateway. The specification of requests is presented in the following sections:
       - "[2.4.1.1. Request for order registration (SOAP)](#_bookmark2)";
       - "[2.4.2.1. Request for order registration (REST)](#_bookmark5)".
    3. In response to the registration request, the Payment gateway returns a unique order identifier in the payment system (in the orderId parameter).
    4. The Merchant redirects the Client to its payment form.
    5. The client chooses to pay via the FPS.
    6. The Merchant sends a request to the Payment gateway to receive a QR code. The request specifications are presented in the following sections:
       - "[2.4.1.2. Request for a QR code (SOAP)](#_bookmark3)";
       - "[2.4.2.2. Request for a QR code (REST)"](#_bookmark6).

7-10. The payment gateway sends the QR code request to the FPS system.

1. The payment gateway returns a response to the QR code request to the merchant's system.
2. The QR code is displayed to the client.
3. The client reads the QR code using specialized software and makes the payment.
4. The payment gateway receives information about the payment made.

15-16. The Merchant requests the payment status from the Payment Gateway. The request specifications are presented in the following sections:

* + "[2.4.1.3. Request for payment status by QR code (SOAP)](#_bookmark4)";
  + ["2.4.2.3. Request for payment status by QR code (REST)](#_bookmark7)".
  1. Payment via the "instant invoice" service

Payment via the "instant invoice" service is possible both on the payment page and on the merchant's side.

In this case, the payment scenarios are similar to those presented in clauses "[2.1. One-phase](#_bookmark0) [payment on the payment page](#_bookmark0)" and ["2.2. One-phase payment with data entry on](#_bookmark1) [the merchant](#_bookmark1)'s side."

When the functionality is enabled, the QR code displayed to the Client is also a link. To make a payment, the Client does not scan the QR code, but clicks on it, after which the online bank opens.

* 1. Description of payment API
     1. SOAP
        1. Request for order registration (SOAP)

The request for order registration is registerOrder

**WSDL (test environment):** https:// tws.egopay.ru/api/ab/webservices/merchant-ws?wsdl

**WSDL (production environment):** https://ecom.alfabank.ru/api/soap/merchant-ws?wsdl

**Request parameters:**

| Name | Type | Mandatory | Description |
| --- | --- | --- | --- |
| merchantOrderNumber | ANS..32 | yes | The order number (identifier) in the store's system is unique for each store within the system |
| description | ANS..598 | no | Description of the order in any form. |
| To enable sending this field to processing, contact technical support. |
| amount | N..12 | yes | Payment amount in minimum currency units |
| currency | N3 | no | The payment currency code is ISO 4217. If not specified, the default value is used. |
| language | A2 | no | The language is encoded in ISO 639-1. If not specified, the default language specified in the store settings will be used. |
| pageView | ANS..20 | no | The value of this parameter determines which pages of the payment interface should be loaded for the client. Possible values are:   * DESKTOP – for pages designed to be displayed on PC screens (pages with names   payment\_<locale>.html and  errors\_<locale>.html  will be searched for in the archive of the payment interface pages);   * MOBILE – for pages designed to be displayed on mobile devices (pages with names mobile\_payment\_<locale>.html and mobile\_errors\_<locale>.html   will be searched for in the archive of the payment interface pages);   * If the store has added arbitrary prefixes to their page file names, value of the desired prefix should be passed in the pageView parameter for the corresponding pages to be loaded. For example, if iphone is passed in the parameter, pages with names iphone\_payment\_<locale>.html and iphone\_error\_<locale>.html   will be searched for in the archive of the payment interface pages.  Where:  locale – the language of the page encoded in ISO 639-1. For example, ru for Russian or en for English.  If the parameter is missing or does not match the format, the default value is pageView=DESKTOP. |
| sessionTimeoutSecs | N...9 | no | The order lifetime in seconds.  If the parameter is not specified, the value specified in the merchant's settings or the default time (1200 seconds = 20 minutes) will be used.  If the request contains the expirationDate parameter, the value of the sessionTimeoutSecs parameter is ignored. |
| bindingId | AN..255 | no | ID of the binding created earlier. It can only be used if the store has permission to work with bindings. If this parameter is passed in this request, it means that:   1. This order can only be paid using a binding; 2. The payer will be redirected to the payment page, where only CVC input is required. |
| expirationDate | ANS | no | Date and time of the order lifetime expiration. Format:  yyyy-MM-dd'T'HH:mm:ss.  If this parameter is not present in the request, then sessionTimeoutSecs is used to determine the time of the order lifetime expiration. |
| returnUrl | ANS..512 | yes | The address to which the user must be redirected in case of successful payment. The address must be specified in full, including the protocol used (for example, **https://test.ru** instead of **test.ru**). Otherwise, the user will be redirected to the address of the following type: **http://<payment\_gateway\_address>/<merchant\_address>.** |
| See also the 3DS payment block - for one-phase and two-phase payments. returnUrl is used in the same way as in step 11 in the specified patterns. |
| failUrl | ANS..512 | no | The address to which the user must be redirected in case of failed payment. The address must be specified in full, including the protocol used (for example, **https://test.ru** instead of **test.ru**). Otherwise, the user will be redirected to the address of the following type: **http://<payment\_gateway\_address>/<merchant\_address>.** |
|  |  |  | See also the 3DS payment block - for one-phase and two-phase payments. failUrl is used in the same way as in step 11 in the specified patterns. |
| dynamicCallbackUrl | ANS..512 | no | This parameter allows you to use the dynamic callback notification sending functionality. All payment-related callbacks enabled for the merchant could be directed to the address passed in this parameter. Payment-related notifications are callback notifications about the following events: successful hold, payment rejected by timeout, cardpresent payment rejected, successful debit, refund, reversal. Notably, payment-unrelated callbacks that are active for the merchant (enabling/disabling a binding, binding creation) will be sent to static callback address. |
|  |  |  | **Note**To use the functionality of dynamic callback notifications the merchant must have the appropriate setting: **Callback type:****Dynamic** (CALLBACK\_TYPE = DYNAMIC). |
|  |  |  |  |
|  |  |  | To receive callback notifications, the merchant must have a permission activation: **Callback operations are allowed.** |
| params | See description |  | Tag containing attributes to pass additional parameters of the merchant.  The fields for additional information and its subsequent storage. To pass N parameters, the request must contain N params tags, where the name attribute contains the name, and the value attribute contains its value:   |  |  |  |  | | --- | --- | --- | --- | | Name | Type | Mandatory | Description | | name | ANS..255 bytes | yes | Name of the additional parameter | | value | ANS..1024 bytes | yes | Value of the additional parameter |   These fields can be passed to the Bank processing system to be subsequently displayed in the registries.\*  This functionality can be enabled upon additional agreement with the bank during the integration period. |
|  |  |  | In case of payments for housing and utilities services, the following additional parameters must be passed:   * payment\_doc\_id – the identifier of the payment document; * order\_num – the order number. |
|  |  |  | If sending notifications to a buyer is set up for a seller, the email address of a buyer must be passed in this tag in the email parameter. |
| clientId | AN..255 | no | The client's number (ID) in the store's system. It is used to implement the functionality of bindings. It can be present if creating bindings is allowed for the store. |
|  |  |  | Specifying this parameter when processing payments with the use of bindings is mandatory. Otherwise, a payment will be failed. |
| merchantLogin | AN..255 | no | To register an order on behalf of a child merchant, specify the merchant login in this parameter. |
| features | ANS..255 | no | A container for feature parameter, where the following values can be specified.  AUTO\_PAYMENT – if the request for order registration initiates auto-payments.  VERIFY – if this parameter is specified, after the request for order registration, the cardholder is to be verified without debiting funds from the cardholder account. Thus, it is possible to pass a zero amount in the request. This verification allows the merchant to ensure that a card belongs to the cardholder and to debit this card in the future without verifying authentication data (CVC, 3-D Secure) when processing subsequent payments. |
|  |  |  | **Specifics of passing the VERIFY value** Even if the payment amount is to be passed in the request, it will not be debited from the buyer's account.After the order has been successfully registered, it is put to the REVERSED (cancelled) status. |
|  |  |  | FORCE\_TDS – use of 3-D Secure is enforced for the payment. If the card does not support 3-D Secure, the transaction will fail.  FORCE\_SSL – use of SSL is enforced for the payment (no 3-D Secure).  FORCE\_FULL\_TDS – after authentication using 3-D Secure, the PaRes status must only be Y, which guarantees successful user authentication. Otherwise, the transaction will fail.  Example of use is given below. |
|  |  |  | <features>  <feature>AUTO\_PAYMENT</feature>  </features> |
| email | ANS..40 | no | Buyer's email address. |
| phone | ANS.12 | no | Buyer's phone number. If the phone includes a country code, the number must begin with a plus sign (+). If the phone is passed without a plus sign (+), the country code should not be specified. Thus, the following options are valid:  • +79998887766;  • 9998887766. |

\* By default the following fields are passed to the bank's processing system:

* orderNumber – order number in the store system;
* description – order description (no more than 99 characters; %, +, carriage return \r, and line feed \n cannot be used).

**Response parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Mandatory** | **Description** |
| orderId | ANS36 | no | The order number in the payment system. It is unique within the system. It is missing if the order registration failed due to an error detailed in ErrorCode. |
| formUrl | AN..512 | no | URL of the payment form to which the client's browser is to be redirected. It is not returned if the order registration failed due to an error detailed in ErrorCode. |
| errorCode | N3 | no | Error code. |
| errorMessage | AN..512 | no | Error description in the language passed in the language parameter in the request. |

**Error Codes** (ErrorCode **field**):

|  |  |
| --- | --- |
| Meaning | Description |
| 0 | Request processing took place without system errors |
| 1 | Invalid order number |
| 1 | Order with given order number is processed already |
| 3 | Currency is not defined |
| 4 | Empty amount |
| 4 | The order number cannot be empty |
| 4 | The return URL cannot be empty |
| 5 | A parameter value has been specified incorrectly |
| 5 | Access denied |
| 5 | The user must change password |
| 7 | System error |
| 13 | Using both values, Features FORCE\_TDS/FORCE\_SSL and AUTO\_PAYMENT, is not allowed |
| 13 | The Merchant does not have the permission to process auto-payments |
| 13 | The Merchant does not have the permission to process verification payments |
| 14 | Features are specified incorrectly |

Request example:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mer="https://ecom.alfabank.ru/api/soap/sbpc2b-ws">

<soapenv:Header/>

<soapenv:Body>

<mer:registerOrder>

<order merchantOrderNumber="78ds901234567890" description=" " amount="15000" currency=" " language=" " pageView="MOBILE" sessionTimeoutSecs=" " bindingId=" " expirationDate="2014-09-08T14:14:14">

<returnUrl>http://yoursite.com</returnUrl>

<params name="param1" value="valueParam1"/>

<params name="param2" value="valueParam2"/>

<clientId>666</clientId>

<merchantLogin> </merchantLogin>

<features>

<feature>AUTO\_PAYMENT</feature>

</features>

</order>

</mer:registerOrder>

</soapenv:Body>

</soapenv:Envelope>

Response example:

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<ns1:registerOrderResponse xmlns:ns1="https://ecom.alfabank.ru/api/soap/sbpc2b-

ws">

<return orderId="05fcbc62-7ee6-4f1a-b3d5-6ca41a982283" errorCode="0" errorMessage="Success">

<formUrl> https://server/application\_context/mobile\_payment\_ru.html?mdOrder=05fcbc62- 7ee6-4f1a-b3d5-6ca41a982283 </formUrl>

</return>

</ns1:registerOrderResponse>

</soap:Body>

</soap:Envelope>

* + - 1. Request for a QR code (SOAP)

WSDL (test environment): https://tws.egopay.ru/api/ab/webservices/sbpc2b-ws?wsdl

WSDL (production environment): https://ecom.alfabank.ru/api/soap/sbpc2b-ws?wsdl The getDynamicQr request is used to get the QR code.

**Request parameter**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Mandatory** | **Description** |
| mdOrder | ANS..36 | Yes | The order number in the payment gateway system. |
| qrHeight | Line | No | The height of the QR code in pixels. Specify if renderedQR  is required. Minimum value: 10. Maximum value: 1000 |
| qrWidth | Line | No | The width of the QR code. Specify if renderedQR is required. Minimum value: 10. Maximum value: 1000. |
| qrFormat | ANS\* | No | Possible values are:  matrix - returns a matrix of zeros and ones; image - returns the image to base64. |

**Response parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Mandatory** | **Description** |
| errorCode | ANS..3 | No | Error code. |
| errorMessage | AN..512 | No | Error description. |
| payload | Line | No | The contents of the QR code registered in the FPS. Present if the qrStatus value is STARTED. |
| qrId | Line | No | QR code ID. |
| qrStatus | Line | No | Status of the QR\_code request. The following values are possible.   * STARTED - QR code generated; * CONFIRMED - the order is accepted for payment; * REJECTED - payment rejected; * REJECTED\_BY\_USER - payment by QR code rejected by the merchant; * ACCEPTED - the order is paid-up. |
| renderedQr |  | No | QR code in PNG format, encoded in Base64.  Present if qrHeight and qrWidth were specified in the request, and the qrStatus value is STARTED. |

**Request format**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mer="https://ecom.alfabank.ru/api/soap/sbpc2b-ws">

<soapenv:Header/>

<soapenv:Body>

<mer:getDynamicQr>

<request>

<mdOrder>string</mdOrder>

<qrHeight>40</qrHeight>

<qrWidth>40</qrWidth>

<qrFormat>image</qrFormat>

</request>

</mer:getDynamicQr>

</soapenv:Body>

</soapenv:Envelope>

**Response format**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<ns1:getDynamicQrResponse xmlns:ns1="https://ecom.alfabank.ru/api/soap/sbpc2b-

ws">

<return>

<errorMessage>string</errorMessage>

<payload>string</payload>

<qrId>string</qrId>

<qrStatus>STARTED</qrStatus>

<renderedQr>string</renderedQr>

</return>

</ns1:getDynamicQrResponse>

</soap:Body>

</soap:Envelope>

* + - 1. Request for payment status by QR code (SOAP)

**WSDL (test environment):** https:// tws.egopay.ru/api/ab/webservices/sbpc2b-ws?wsdl

**WSDL (production environment):** https://ecom.alfabank.ru/api/soap/sbpc2b-ws?wsdl

The getQrStatus request is used to get the order status in the FPS.

**Request parameter**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Mandatory** | **Description** |
| mdOrder | ANS..36 | Yes | The order number in the payment gateway system. |
| qrId | Line | Yes | QR code ID. |

**Response parameters**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Parameter** | | **Type** | | **Mandatory** | | **Description** | |
| errorCode | | ANS..3 | | No | | Error code. | |
| errorMessage | | AN..512 | | No | | Error description. | |
| payload | | Line | | No | | The contents of the QR code registered in the FPS. Present if the qrStatus value is STARTED. | |
| qrStatus | | Line | | No | | Status of the QR\_code request. The following values are possible.   * STARTED - QR code generated; * CONFIRMED - the order is accepted for payment; * REJECTED - payment rejected; * REJECTED\_BY\_USER - payment by QR code rejected by the merchant; * ACCEPTED - the order is paid-up. | |
| qrType | | Line | | No | | Type of QR code:   * STATIC - static; * DYNAMIC - dynamic.   Currently, the DYNAMIC value is always returned. | |
| transactionState | | Line | | No | | Order status:   * CREATED - created; DECLINED - declined; * DEPOSITED - paid-up. | |
| transactionStateExtension | | Line | | No | | Advanced order status:   * SBP\_C2B\_STARTED - the order is created; * SBP\_C2B\_CONFIRMED - the order is confirmed; * SBP\_C2B\_PENDING - the order is waiting for processing. | |

**Request format**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mer="https://ecom.alfabank.ru/api/soap/sbpc2b-ws">

<soapenv:Header/>

<soapenv:Body>

<mer:getQrStatus>

<request>

<mdOrder>string</qrId>

<qrId>string</qrStatus>

</request>

</mer:getQrStatus>

</soapenv:Body>

</soapenv:Envelope>

**Response format**

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<ns1:getQrStatusResponse xmlns:ns1="http://test.ru/webservices/merchant">

<return>

<errorCode>string</errorCode>

<errorMessage>string</errorMessage>

<payload>string</payload>

<qrStatus>STARTED</qrStatus>

<qrType>STATIC</qrType>

<transactionState>CREATED</transactionState>

<transactionStateExtension>SBP\_C2B\_STARTED</transactionStateExtension>

</return>

</ns1:getQrStatusResponse>

</soap:Body>

</soap:Envelope>

* + 1. REST
       1. Request for order registration (REST)

The register.do request is used to register an order.

**URL (test environment):** https:// tws.egopay.ru/api/ab/rest/register.do

**URL (production environment):** https://ecom.alfabank.ru/api/rest/register.do

**Request parameters:**

| **Name** | **Type** | **Mandatory** | **Description** |
| --- | --- | --- | --- |
| userName | AN..30 | no (either a login and password, or a token needs to be specified) | AN..30 |
| password | AN..30 | no (either a login and password, or a token needs to be specified) | AN..30 |
| token | AN..30 | no (either a login and password, or a token needs to be specified) | ANS..30 |
| orderNumber | AN..32 | yes | The order number (identifier) in the store's system is unique for each store within the system |
| amount | N..12 | yes | Payment amount in kopecks (or cents) |
| currency | N3 | no | The payment currency code is ISO 4217. If not specified, the default value is used. |
| returnUrl | ANS..512 | yes | The address to which the user must be redirected in case of successful payment. The address must be specified in full, including the protocol used (for example, **https://test.ru** instead of **test.ru**). Otherwise, the user will be redirected to the address of the following type: **http://<payment\_gateway\_address>/<merchant\_address>.** |
|  |  |  | See also the 3DS payment block - for one-phase and two-phase payments. returnUrl is used in the same way as in step 11 in the specified patterns. |
| failUrl | ANS..512 | no | The address to which the user must be redirected in case of failed payment. The address must be specified in full, including the protocol used (for example, **https://test.ru** instead of **test.ru**). Otherwise, the user will be redirected to the address of the following type: **http://<payment\_gateway\_address>/<merchant\_address>.** |
|  |  |  | See also the 3DS payment block - for one-phase and two-phase payments. failUrl is used in the same way as in step 11 in the specified patterns. |
| dynamicCallbackUrl | ANS..512 | no | This parameter allows you to use the dynamic callback notification sending functionality. All payment-related callbacks enabled for the merchant could be directed to the address passed in this parameter. Payment-related notifications are callback notifications about the following events: successful hold, payment rejected by timeout, cardpresent payment rejected, successful debit, refund, reversal. Notably, payment-unrelated callbacks that are active for the merchant (enabling/disabling a binding, binding creation) will be sent to static callback address. |
|  |  |  | **Note**To use the functionality of dynamic callback notifications the merchant must have the appropriate setting: **Callback type: Dynamic** (CALLBACK\_TYPE = DYNAMIC). |
|  |  |  |  |
|  |  |  | To receive callback notifications, the merchant must have a permission activation: **Callback operations are allowed.** |
| description | ANS..598 | no | Description of the order in any form. |
|  |  |  | To enable sending this field to processing, contact technical support. |
| language | A2 | no | The language is encoded in ISO 639-1. If not specified, the default language specified in the store settings will be used. |
| pageView | ANS..20 | no | The value of this parameter determines which pages of the payment interface should be loaded for the client. Possible values are:   * DESKTOP – for pages designed to be displayed on PC screens (pages with names payment\_<locale>.html and errors\_<locale>.html will be searched for in the archive of the payment interface pages); * MOBILE – for pages designed to be displayed on mobile devices (pages with names * mobile\_payment\_<locale>.html and mobile\_errors\_<locale>.html   will be searched for in the archive of the payment interface pages);   * If the store has added arbitrary prefixes to their page file names, value of the desired prefix should be passed   in the pageView parameter for the corresponding pages to be loaded. For example, if iphone is passed in the parameter, pages with names  iphone\_payment\_<locale>.html and iphone\_error\_<locale>.html  will be searched for in the archive of the payment interface pages.  Where:  locale – the language of the page encoded in ISO 639-1. For example, ru for Russian or en for English.  If the parameter is missing or does not match the format, the default value is pageView=DESKTOP. |
| clientId | AN..255 | no | The client's number (ID) in the store's system. It is used to implement the functionality of bindings. It can be present if creating bindings is allowed for the store. |
|  |  |  | Specifying this parameter when processing payments with the use of bindings is mandatory. Otherwise, a payment will be failed. |
| merchantLogin | AN..255 | no | To register an order on behalf of a child merchant, specify the merchant login in this parameter. |
| jsonParams | See description | no | A block for passing additional merchant parameters. The fields for additional information and its subsequent storage are passed as:  {"<name1>":"<value1>",...,"<nameN>":"<valueN>"},  These fields can be passed to the Bank processing system to be subsequently displayed in the registries.\*  This functionality can be enabled upon additional agreement with the bank during the integration period.  If sending notifications to a buyer is set up for a seller, the email address of a buyer must be passed in this block in the email parameter.  Data type   |  |  | | --- | --- | | **Name** | **Type** | | name | ANS..255 bytes | | value | ANS..1024 bytes | |
|  |  |  | In case of payments for housing and utilities services, the following additional parameters must be passed:   * payment\_doc\_id – the identifier of the payment document; * order\_num – the order number. |
| sessionTimeoutSecs | N...9 | no | The order lifetime in seconds.  If the parameter is not specified, the value specified in the merchant's settings or the default time (1200 seconds = 20 minutes) will be used.  If the request contains the expirationDate parameter, the value of the sessionTimeoutSecs parameter is ignored. |
| expirationDate | ANS | no | Date and time of the order lifetime expiration. Format:  yyyy-MM-ddTHH:mm:ss.  If this parameter is not present in the request, then sessionTimeoutSecs is used to determine the time of the order lifetime expiration. |
| bindingId | AN..255 | no | ID of the binding created earlier. It can only be used if the store has permission to work with bindings. If this parameter is passed in this request, it means that:   1. This order can only be paid using a binding; 2. The payer will be redirected to the payment page, where only CVC input is required. |
| features | ANS..255 | no | The following values can be used.  AUTO\_PAYMENT – The payment is made without authentication of the cardholder (without CVC and 3D-Secure). In order to make such payments, the seller must have the appropriate permits.  VERIFY – if this parameter is specified, after the request for order registration, the cardholder is to be verified without debiting funds from the cardholder account. Thus, it is possible to pass a zero amount in the request. This verification allows the merchant to ensure that a card belongs to the cardholder and to debit this card in the future without verifying authentication data (CVC, 3-D Secure) when processing subsequent payments. |
|  |  |  | **Specifics of passing the VERIFY value** Even if the payment amount is to be passed in the request, it will not be debited from the buyer's account.After the order has been successfully registered, it is put to the REVERSED (cancelled) status. |
|  |  |  | FORCE\_TDS – use of 3-D Secure is enforced for the payment. If the card does not support 3-D Secure, the transaction will fail.  FORCE\_SSL – use of SSL is enforced for the payment (no 3-D Secure).  FORCE\_FULL\_TDS – after authentication using 3-D Secure, the PaRes status must only be Y, which guarantees successful user authentication. Otherwise, the transaction will fail. |
| email | ANS..40 | no | Buyer's email address. |
| phone | ANS.12 | no | Buyer's phone number. If the phone includes a country code, the number must begin with a plus sign (+). If the phone is passed without a plus sign (+), the country code should not be specified. Thus, the following options are valid:  • +79998887766;  • 9998887766. |



\* By default the following fields are passed to the bank's processing system:

* orderNumber – order number in the store system;
* description – order description (no more than 99 characters;

%, +, carriage return \r, and line feed \n cannot be used).

**Response parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Mandatory** | **Description** |
| orderId | ANS36 | No | The order number in the payment system. It is unique within the system. It is missing if the order registration failed due to an error detailed in ErrorCode. |
| formUrl | AN..512 | No | URL of the payment form to which the client's browser is to be redirected. It is not returned if the order registration failed due to an error detailed in ErrorCode. |
| errorCode | N3 | No | Error code. |
| errorMessage | AN..512 | no | Error description in the language passed in the language parameter in the request. |

**Error Codes** (ErrorCode **field**):

|  |  |
| --- | --- |
| **Meaning** | **Description** |
| 0 | Request processing took place without system errors |
| 1 | Order with given order number is processed already |
| 1 | An order with this number was registered, but was not paid for |
| 1 | Invalid order number |
| 3 | Currency is not defined |
| 4 | The order number cannot be empty |
| 4 | The merchant's name cannot be empty |
| 4 | Empty amount |
| 4 | The return URL cannot be empty |
| 4 | The password cannot be empty |
| 5 | The seller's login is incorrect |
| 5 | Incorrect amount |
| 5 | Incorrect 'Language' parameter |
| 5 | Access denied |
| 5 | The user must change password |
| 5 | Access denied |
| 5 | [jsonParams] is invalid |
| 7 | System error |
| 13 | Using both values, Features FORCE\_TDS/FORCE\_SSL and AUTO\_PAYMENT, is not allowed |
| 13 | The Merchant does not have the permission to process auto-payments |
| 13 | The Merchant does not have the permission to process verification payments |
| 14 | Features are specified incorrectly |

**Example of a POST request:**

https:// tws.egopay.ru/api/ab/rest/register.do?userName=login- api&password=login\*%3F1&amount=100&currency=810&language=ru&orderNumbe r=87654321&returnUrl=http://yoursite.com&pageView=DESKTOP&jsonParams={ "param1":"value1","param2":"value2"}&expirationDate=2014-09- 08T14:14:14

**Response example:**

{"orderId":"70906e55-7114-41d6-8332-

4609dc6590f4","formUrl":"https://server/application\_context//merchants

/test/payment\_ru.html?mdOrder=70906e55-7114-41d6-8332-4609dc6590f4"}

* + - 1. Request for a QR code (REST)

The get.do request is used to get a QR code.

**URL (test environment):** https:// tws.egopay.ru/api/ab/rest/sbp/c2b/qr/dynamic/get.do

**URL (production environment):** https://ecom.alfabank.ru/api/rest/sbp/c2b/qr/dynamic/get.do

**Request format:** POST

**Content-type:** application/x-www-form-urlencoded

**Request parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Mandatory** | **Description** |
| *Parameters in the request body, format x-www-form-urlencoded:* | | | |
| userName | AN..30 | Yes | The store's login received when connecting. |
| password | AN..30 | Yes | The store's password received when connecting. |
| mdOrder | ANS..36 | Yes | The order number in the payment gateway system. |
| qrHeight | Line | No | The height of the QR code in pixels. Specify if required  renderedQR. Minimum value: 10. Maximum value: 1000 |
| qrWidth | Line | No | The width of the QR code. Specify if renderedQR is required. Minimum value: 10. Maximum value: 1000. |
| qrFormat | ANS\* | No | Possible values are:  matrix - returns a matrix of zeros and ones; image - returns the image to base64. |

**Response parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Mandatory** | **Description** |
| errorCode | ANS..3 | No | Error code. |
| errorMessage | AN..512 | No | Error description. |
| payload | Line | No | The contents of the QR code registered in the FPS.  Present if the qrStatus value —  STARTED. |
| qrId | Line | No | QR code ID. |
| qrStatus | Line | No | Status of the QR\_code request. The following values are possible.   * STARTED - QR code generated; * CONFIRMED - the order is accepted for payment; * REJECTED - payment rejected; * REJECTED\_BY\_USER - payment by QR code rejected by the merchant; * ACCEPTED - the order is paid-up. |

**Request format**

POST https:// tws.egopay.ru/api/ab/rest/sbp/c2b/qr/dynamic/get.do&? userName=userName&password=password&mdOrder=04888d6f-7920-7531-8332- 8de901efddd0&qrFormat=image

**Response format**

{

"qrId": "54d14bae6f7f4a73929308e9afa5915d", "payload":

"https://qr.nspk.ru/54d14bae6f7f4a73929308e9afa5915d?type=02&bank=6000000000 15&sum=13000&cur=RUB&crc=464g",

"qrStatus": "STARTED"

}

* + - 1. Request for payment status by QR code (REST)

The status.do request is used to get the request status in the FPS.

**URL (test environment):** https:// tws.egopay.ru/api/ab/rest/sbp/c2b/qr/status.do

**URL (production environment):** https://ecom.alfabank.ru/api/rest/sbp/c2b/qr/status.do Request format: POST

**Content-type:** application/x-www-form-urlencoded

**Request parameter**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Parameter** | | **Type** | | **Mandatory** | | **Description** | |
| *Parameters in the request body, format x-www-form-urlencoded:* | | | | | | | |
| userName | | AN..30 | | Yes | | The store's login received when connecting. | |
| password | | AN..30 | | Yes | | The store's password received when connecting. | |
| mdOrder | | ANS..36 | | Yes | | The order number in the payment gateway system. | |
| qrId | | Line | | Yes | | QR code ID. | |

**Response parameters**

| **Parameter** | **Type** | **Mandatory** | **Description** |
| --- | --- | --- | --- |
| errorCode | ANS..3 | No | Error code. |
| errorMessage | AN..512 | No | Error description. |
| qrStatus | Line | No | Status of the QR\_code request. The following values are possible.   * STARTED - QR code generated; * CONFIRMED - the order is accepted for payment; * REJECTED - payment rejected; * REJECTED\_BY\_USER - payment by QR code rejected by the merchant; * ACCEPTED - the order is paid-up. |
| qrType | Line | No | Type of QR code:   * STATIC - static; * DYNAMIC - dynamic.   Currently, the DYNAMIC value is always returned. |
| transactionState | Line | No | Order status:   * CREATED - created; DECLINED - declined; * DEPOSITED - paid-up. |

**Request format**

POST https:// tws.egopay.ru/api/ab/rest/sbp/c2b/qr/status.do? userName=userName&password=password&mdOrder=04888d6f-7920-7531-8332- 8de901efddd0&qrId=3946c0c02d1042f7b7e63cc0f1b52a95

**Response format**

{

"qrType": "DYNAMIC",

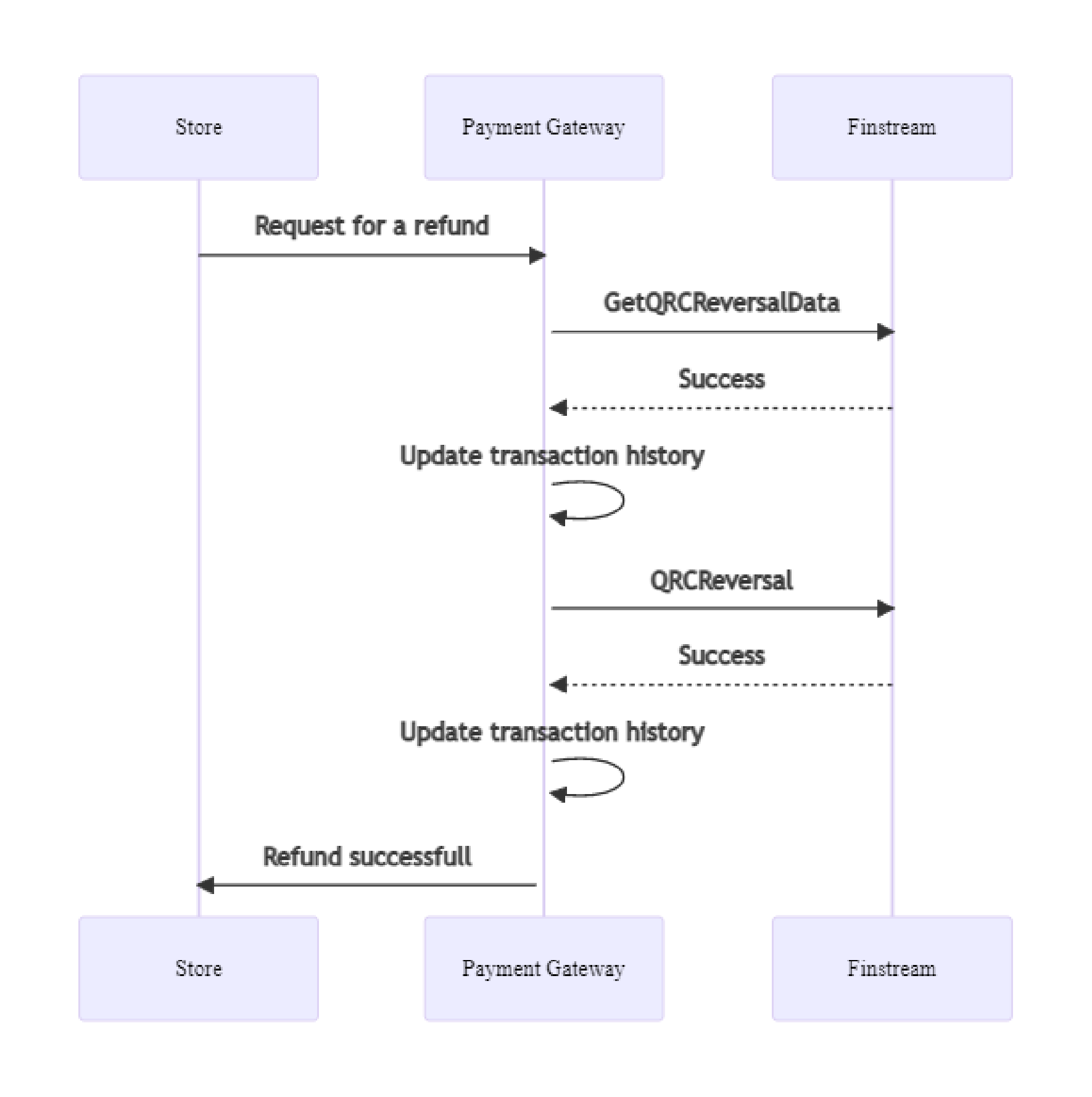
"qrStatus": "ACCEPTED", "transactionState": "DEPOSITED"

}

1. Refund

Note: Refunds are possible for Merchants who have been assigned the appropriate permissions in the Payment gateway.

* 1. Refund via API
     1. Scenario of a refund via API



Description:

* + - 1. The Merchant sends a request for a refund to the Payment gateway. The specification of requests is presented in the following sections:
         * ["3.1.2.1. Refund request (SOAP)";](#_bookmark8)
         * ["3.1.2.2. Refund request (REST)".](#_bookmark9)
      2. The Payment gateway sends to the FPS system a request for the possibility of a refund.
      3. After receiving the response, the transaction history is updated in the Payment gateway.
      4. In case of a successful response to the request for a refund, the Payment gateway sends a refund request to the FPS system.
      5. After receiving the response, the transaction history is updated in the Payment gateway.
      6. The Payment gateway returns the result of the refund operation.

If an error is received at Step 2 or Step 4, then after updating the transaction history, the Payment gateway sends errorCode=7 - "System error" to the Merchant's system.

* + 1. Description of the API for refund
       1. Refund request (SOAP)

The refundOrder request is used for the refund.

**WSDL (test environment):** https://tws.egopay.ru/api/ab/webservices/merchant-ws?wsdl

**WSDL (production environment):** https://ecom.alfabank.ru/api/soap/merchant-ws?wsdl

Upon this request, the funds for the specified order will be returned to the payer. The request will end with an error if the funds for this order have not been debited. The system enables to return funds more than once, but for a total amount not exceeding the initial debit amount.

|  |
| --- |
| When processing a refund for housing and utilities services, only a full refund is available. |

To process the refund operation, one must have the appropriate rights in the system.

**Request parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Mandatory** | **Description** |
| orderId | ANS36 | yes | The order number in the payment system. It is unique within the system. |
| refundAmount | N..12 | yes | The refund amount in the order currency. May be less than or equal to the remnant in the order. |
| language | A2 | no | The language is encoded in ISO 639-1. If not specified, it is assumed that the language is Russian. The error message will be returned in this language. |

**Response parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Mandatory** | **Description** |
| errorCode | N3 | no | Error code. |
| errorMessage | AN..512 | no | Error description in the language passed in the Language parameter in the request. |

Error Codes (ErrorCode field):

|  |  |
| --- | --- |
| **Meaning** | **Description** |
| 0 | Request processing took place without system errors |
| 5 | Access denied |
| 5 | The user must change password |
| 5 | [orderId] not set |
| 5 | Incorrect amount |
| 6 | Invalid order number |
| 7 | The payment must be in the correct state |
| 7 | The refund amount exceeds the amount debited |
| 7 | System error |

**Request example:**

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:mer="https://ecom.alfabank.ru/api/soap/sbpc2b-ws">

<soapenv:Header/>

<soapenv:Body>

<mer:refundOrder>

<order language="ru" orderId="4302d369-a5e8-4432-a5e5- 42acfab52c86" refundAmount="20000">

<!--Zero or more repetitions:-->

<params name=" " value=" "/>

</order>

</mer:refundOrder>

</soapenv:Body>

</soapenv:Envelope>

**Response example:**

<soap:Envelope xmlns:soap=["http://schemas.xmlsoap.org/soap/envelope/](http://schemas.xmlsoap.org/soap/envelope/)">

<soap:Body>

<ns1:refundOrderResponse xmlns:ns1="https://ecom.alfabank.ru/api/soap/sbpc2b-

ws">

<return ErrorCode="7" ErrorMessage="The refund amount exceeds the amount debited"/>

</ns1:refundOrderResponse>

</soap:Body>

</soap:Envelope>

* + - 1. Refund request (REST)

The refund.do request is used for the refund.

**URL (test environment):** https:// tws.egopay.ru/api/ab/rest/refund.do

**URL (production environment):** https://ecom.alfabank.ru/api/rest/refund.do

Upon this request, the funds for the specified order will be returned to the payer. The request will end with an error if the funds for this order have not been debited. The system enables to return funds more than once, but for a total amount not exceeding the initial debit amount.



When processing a refund for housing and utilities services, only a full refund is available.

To process the refund operation, one must have the appropriate rights in the system.

**Request parameters:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | | **Compulsoriness** | | **Description** | | |
| userName | AN..30 | | yes | | The store's login received when connecting | | |
| password | AN..30 | | yes | | The store's password received when connecting | | |
| orderId | ANS36 | | yes | | The order number in the payment system. It is unique within the system. | | |
| amount | | N..12 | yes | Payment amount in kopecks (or cents) | | | |
| language | | A2 | no | The language is encoded in ISO 639-1. If not specified, it is assumed that the language is Russian. The error message will be returned in this language. | | | |
| jsonParams | | See description | no | The fields for additional information and its subsequent storage, such as  {"param":value,"param2":value 2}. These fields can be passed to the Bank processing system to be subsequently displayed in the registries.\* This functionality can be enabled upon additional agreement with the bank during the integration period.  Data type | | | |
|  | | **Name** | **Type** |
| name | | | ANS..255 bytes |
| value | | | ANS..1024 bytes |

**Response parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Compulsoriness** | **Description** |
| errorCode | N3 | No | Error code. |
| errorMessage | AN..512 | No | Description of the error in the language. |

**Error Codes** (ErrorCode **field**):

Classification:

|  |  |
| --- | --- |
| **Meaning** | **Description** |
| 0 | Request processing took place without system errors |
| 5 | Incorrect value of a request parameter |
| 6 | Unregistered OrderId |
| 7 | System error |

Decoding:

|  |  |
| --- | --- |
| **Meaning** | **Description** |
| 0 | Request processing took place without system errors |
| 5 | Access denied |
| 5 | The user must change password |
| 5 | [orderId] not set |
| 6 | Invalid order number |
| 7 | The payment must be in the correct state |
| 7 | Incorrect deposit amount (less than one ruble) |
| 7 | System error |

**Example of a POST request:**

https:// tws.egopay.ru/api/ab/rest/refund.do?userName=login- api&password=login\*%3F1&amount=500&currency=810&language=ru&orderId=5e 97e3fd-1d20-4b4b-a542-f5995f5e8208

**Response example:**

{"errorCode":0}

* 1. Refund via PA

Refund after payment via the FPS is possible via PA.

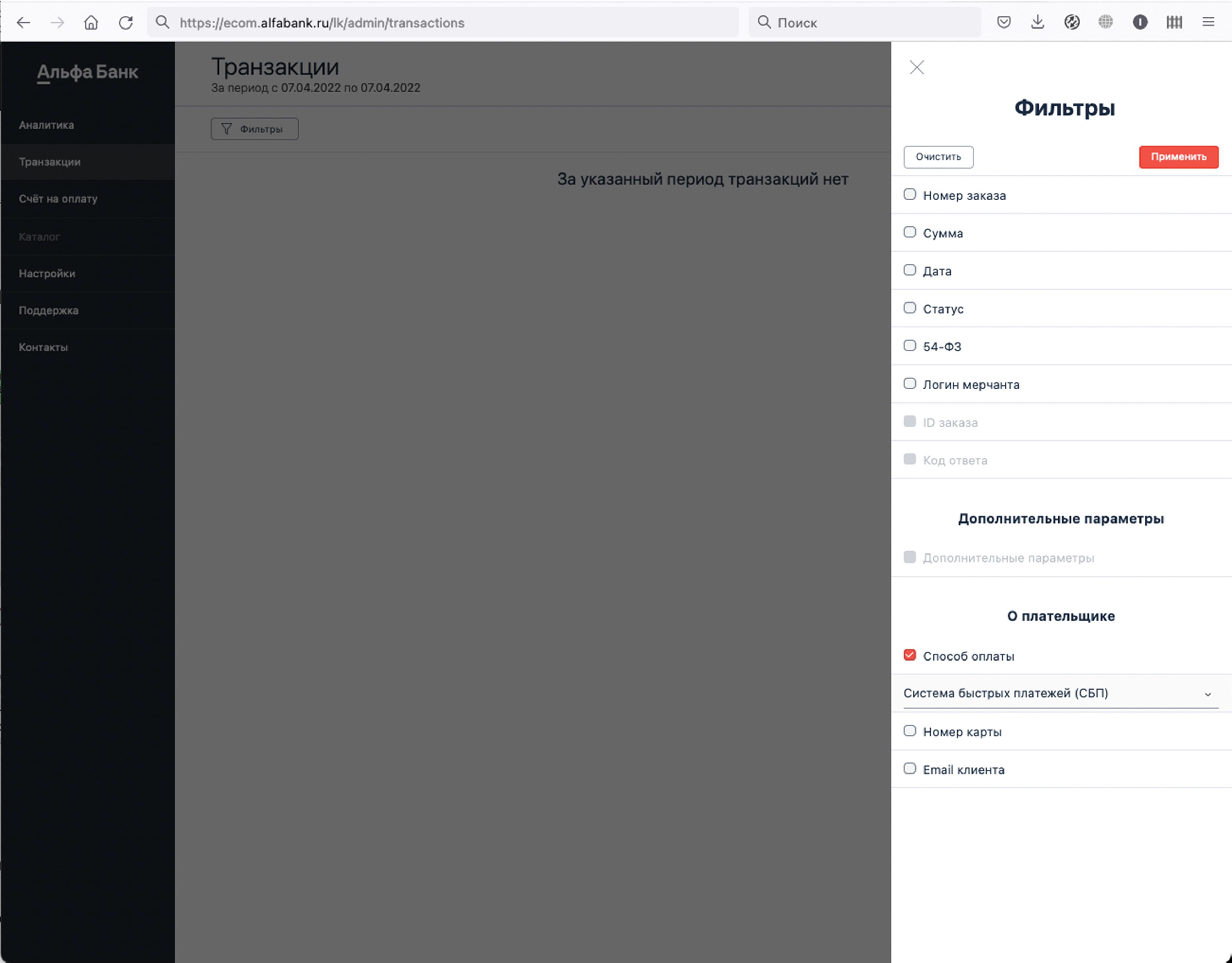
### A description of the refund process via PA is provided in the document "[Instructions for](https://ecom.alfabank.ru/assets/instructions/MPortal3.pdf#page%3D31) [working with the Personal Area](https://ecom.alfabank.ru/assets/instructions/MPortal3.pdf#page%3D31)" (see cl. "7.3. Cancellation and refund for the order").

The search in the PA for orders paid through the FPS is carried out as described in cl. ["7.2.](https://ecom.alfabank.ru/assets/instructions/SBP_C2B.pdf#page%3D32)  [Order completion](https://ecom.alfabank.ru/assets/instructions/SBP_C2B.pdf#page%3D32)" below.

1. Viewing in PA of orders paid via the FPS
   1. Search for orders

To search for orders paid via the FPS in the PA, select "Alpha FPS" in the "Payment Way" field on the order filter panel:

For more information about the order filter, see the document ["Instructions for working with the Personal Area"](https://ecom.alfabank.ru/assets/instructions/MPortal3.pdf#page%3D27) (cl. "7.1 Search for orders using the Filter").



* 1. Order history

The FPS operations are recorded in the order history.

### For more information about viewing the order history in the PA, see the document "[Instructions for working](https://ecom.alfabank.ru/assets/instructions/MPortal3.pdf#page%3D32) [with the Personal Area](https://ecom.alfabank.ru/assets/instructions/MPortal3.pdf#page%3D32)" (cl. "7.4 Viewing transaction details").

1. Fiscalization of transactions

In case of successful payment, a fiscal receipt is issued.

In case of a successful refund, a fiscal receipt for the refund is issued.

Setting up fiscalization <https://ecom.alfabank.ru/faq/fiscal.html>

1. Stubbing mode on the testing fixture

Payment on test payment page:

1. It is necessary to choose payment via the FPS;
2. A QR code will be displayed on the screen.
3. After a few seconds, the payment result will be shown on the screen.

The payment result depends on the order amount:

* the amount is less than 500 rubles – the payment is successful.
* the amount is more than 500 rubles – the payment is unsuccessful.

If a non-standard payment page is used, the button for payment via the FPS may not be available. In this case, you should contact the Bank.