

# Possible interaction schemes

In accordance with Russian law, merchants must send electronic versions of checks to the fiscal data operator via the Internet. There is an opportunity to simplify this procedure: you will send the necessary data in requests, and Sberbank will register receipts on POS equipment.

- [Interaction schemes with standard payment](#) (receipts are registered by Merchant)
- [Interaction schemes with receipt registration on POS equipment](#) (receipts are registered by Sberbank)
  - [Standard API](#)
  - [Apple Pay](#)
  - [Google Pay](#)
  - [Samsung Pay](#)
- [Payment with the provision of the possibility of Online Lending](#) (scenarios of interaction when for payment with providing the buyer with the opportunity to get a loan)
- [Interaction schemes with payment by binding](#)

## Regular payment

### Payment with card detail input on the payment page

	One-phase payment	Two-phase payment
1	The payer chooses goods or services on the merchant website and picks bank card as a payment option.	
2	<p>The merchant sends an order registration request to the payment gateway:</p> <ul style="list-style-type: none"><li>• <a href="#">REST interface (register.do)</a>;</li><li>• <a href="#">WSDL interface (registerOrder)</a>.</li></ul> <p>Among others, the following parameters are passed in the request:</p> <ul style="list-style-type: none"><li>• returnUrl – the URL to which the payer will be redirected if payment is successful;</li><li>• failUrl – the URL to which the payer will be redirected if payment is unsuccessful.</li></ul>	<p>The merchant sends an order registration with pre-authorization request to the payment gateway:</p> <ul style="list-style-type: none"><li>• <a href="#">REST interface (registerPreAuth.do)</a>;</li><li>• <a href="#">WSDL interface (registerOrderPreAuth)</a>.</li></ul> <p>Among others, the following parameters are passed in the request:</p> <ul style="list-style-type: none"><li>• amount – amount to be debited;</li><li>• orderNumber – order number in the store system;</li><li>• returnUrl – the URL to which the payer will be redirected if payment is successful;</li><li>• failUrl – the URL to which the payer will be redirected if payment is unsuccessful.</li></ul>
3	<p>Payment gateway returns a response which, among others, includes the following parameters:</p> <ul style="list-style-type: none"><li>• orderId – unique identifier of the order in the payment system;</li><li>• returnUrl – URL of the form for entering card details (the payment form).</li></ul>	
4	Merchant passes to payer the URL specified in returnUrl parameter of the response.	

	One-phase payment	Two-phase payment
5	Payer's browser opens a form for entering card details. The payer fills in the received form and sends the data to the payment gateway server.	
6	<p>Further actions depend on whether the customer's card supports 3-D Secure or not.</p> <ul style="list-style-type: none"> <li>• 3-D Secure not supported – go to next step;</li> <li>• 3-D Secure supported – the Payment Gateway redirects the payer to authentication form (in most cases, this is a page where SMS code must be entered) stored on the Access Control Server which belongs to issuing bank. If authentication is successful, next step is carried out.</li> </ul>	
7	Payment gateway debits funds from payer's account.	Payment gateway holds (reserves) funds on payer's account.
8	After the payment is carried out, the Payment Gateway redirects the Customer to the return URL (the return URL that is passed by the Store in request for order registration).	
9	Customer's browser requests from the Store the payment result page.	
10	<p>The merchant requests from payment gateway the order payment status by passing a unique identifier that was received upon order registration in orderId parameter:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do)</a>;</li> <li>• <a href="#">WSDL interface (getOrderStatusExtended)</a>.</li> </ul>	
11	The payment gateway returns payment status and the merchant passes to the customer's browser the page with payment result.	
12	Not applicable	<p>To debit funds from the Customer's account, the Merchant needs to send to the Payment Gateway an order completion request:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (deposit.do)</a>;</li> <li>• <a href="#">WSDL interface (depositOrder)</a>.</li> </ul>
13	Not applicable	<p>The payment gateway returns the result of processing the request. The order status is not returned. To get the status of the order, an order status request should be sent to the gateway:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do)</a>;</li> <li>• <a href="#">WSDL interface (getOrderStatusExtended)</a>.</li> </ul>

There is an option to set up [callback notifications](#) about order status instead of sending the `getOrderStatusExtended` request. These notifications will be sent to you automatically after the order status changes.

## Payment reversal

Reversal of a payment for an order is available to Merchants provided that they have the corresponding permissions and approval of Sberbank.

- With one-phase payments, payment reversal is possible for orders in the state **Deposited**, but only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.

- In case of two-phase payments, a payment reversal can be processed for orders in the **Approved** status.

The table below shows how the merchant and the payment gateway interact when a payment is reversed.

1	<p>The merchant makes a request to reverse the payment:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (reverse.do);</a></li> <li>• <a href="#">WSDL interface (reverseOrder).</a></li> </ul>
2	<p>After receiving a successful response from the Payment Gateway, the Merchant makes a request for order status:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do);</a></li> <li>• <a href="#">WSDL interface (getOrderStatusExtended).</a></li> </ul>

There is an option to set up [callback notifications](#) about order status instead of sending the getOrderStatusExtended request. These notifications will be sent to you automatically after the order status changes.

## Refunds to the payer

The Payment Gateway allows full or partial refunds for paid orders.

### callback instead of getorderstatusextended

There is an option to set up [callback notifications](#) about order status instead of sending the getOrderStatusExtended request. These notifications will be sent to you automatically after the order status changes.

### getreceiptstatus

To get information about a sales receipt (for example, to then pass this data to a customer) use the [getReceiptStatus.do](#) request. Currently, this request can only be made through the REST interface.

### recommended wordpress version

Recommended version of WordPress for working with the Sberbank payment plugin: not lower than 4.8.3.

### download woocommerce

You can download the required version of WooCommerce on [github.com](https://github.com/woocommerce/woocommerce) (link leads to WooCommerce version history).

### **woocommerce compatibility**

The WooCommerce plugin is not compatible with all WordPress themes. Here, the [Storefront](#) theme is used as example. This theme is designed by WooCommerce developers.

### **meanings of terms**

Meanings of terms highlighted on this page in **bold** font are given in the [glossary](#).

### **types of payments**

One-phase payments are preferable if the product or service is provided immediately after payment. Two-phase payments should be used if some time elapses between the buyer's decision to pay and the delivery of the selected product or service.

### **need to check 3dsec**

The need for verification using 3-D Secure depends on the buyer's bank card, as well as on the settings configured for the Merchant upon agreement with Sberbank.

### **reversal before midnight**

In case of one-phase payments, the reversal operation is possible only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.

### **access to personal account**

Regardless of the chosen connection method, you are given access to the [personal account](#) on the payment gateway server.

### **you can use test cards**

You can use [test cards](#) to try out the functionality.

### **access to payment methods**

The available payment acceptance methods depend on settings configured for you. If in doubt, check

with a Sberbank representative which method of accepting payments you can use.

### **passed as clientid**

The value of this parameter goes in the registration request as the `clientId` parameter.

### **need to use utf-8**

The page of your site on which the payment solution will be used must be saved in UTF-8 encoding, and it must also have a meta-tag `<meta name=«viewport» content=«width=device-width»>`.

### **update the links**

The links point to test environment. Once done with testing, update the links so that they point to the production environment.

### **tls is required**

The server on which the content management system is located must support the TLS 1.2 protocol. If in doubt, check with your hosting provider.

### **minimum amount to be debited**

The minimum amount of funds to be debited is 100 rubles.

### **default parameters**

If the cart and fiscalization data are not passed in the request, the default values specified in the personal account are passed to the Fiscal Data Operator (see [setting up fiscalization data](#) for more details).

### **passing a field to processing**

To enable sending this field to the processing system, contact the technical support service.

### **parameter must be specified**

Specifying this parameter when processing payments with the use of bindings is mandatory. Otherwise, a payment will be unsuccessful.

## **fields are passed by default**

By default, the following fields are passed to the Bank processing system:

- `orderNumber` – order number in the store system;
- `description` – order description (no more than 24 characters; %, +, carriage return \r, and line feed \n cannot be used).

## **mandatory to pass either email or phone**

It is mandatory to pass one of the two parameters: `email` or `phone`.

## **parameters are not required if the block is missing**

A parameter included in a block and indicated as mandatory is only mandatory within that block. If the section is optional and missing, then the parameters included in it should not be passed.

## **access to personal account**

Access to personal account is granted regardless of the chosen connection method.

## **access to functions depends on settings**

Functions available in the personal account depend on the configured settings. If in doubt, please contact a Sberbank representative.

## **more details available on the 1c-bitrix website**

For more details on the differences between editions, see the website [«1C-Bitrix»](#).

## **1c-bitrix version used as example**

1C-Bitrix version used for installation and configuration: **1C-Bitrix: Site Manager, Small Business Edition**, version **16.5.4**.

## **terms in the 1c-bitrix interface**

In the 1C-Bitrix interface, a «payment system» is a payment method that a customer can choose, for example: «in cash to a courier», «by bank card», etc. The term «payment method» is used throughout this description.

## 1C-Bitrix settings order

The order of fields and settings may differ depending on the edition of your 1C-Bitrix instance.

## 1C-Bitrix using default tab as example

In this description, the setting is shown using the example of the **Default** tab. If the tabs **Individual** and **Entity** have not been edited before, the changes made will apply to these tabs.

### password if the plugin is in test mode

If the plugin is running in test mode, you should specify the password for the test user account. If the plugin is running in standard (production) mode, you should specify the password for the production user account.

### login if the plugin is in test mode

If the plugin is running in test mode, you should specify the login for the test user account. If the plugin is running in production mode, you should specify the password for the production user account.

### wordpress display order

If you want to change the order in which a payment method is displayed for a customer, click on the icon opposite the required payment method and, while holding the left mouse button, move this payment method to the required position.

### cs-cart compatibility

The payment plugin was tested for compatibility with CS-Cart version 4.6.x.

### terms in the cs-cart interface

In the CS-Cart interface, modules that extend the site's functionality are called «add-ons». In this description, the term «plugin» is used.

### plugin settings only

The description contains only those settings that relate directly to the payment plugin. The rest of the settings are related to the content management system (CMS).

## **terms in the Drupal interface**

In the Drupal interface, add-ons are called «modules». The payment plugin falls into the same category, however the term «plugin» is used in this description.

## **drupal description based on example**

A combined distribution [Commerce Kickstart](#) version 7.x-2.45 was used for the purposes of this description. This distribution combines Drupal version 7.54 and Drupal Commerce version 7.x-1.13.

## **opencart compatibility**

The plugin was tested on OpenCart/ocStore version 2.3.0.2.

## **the location of the log files**

The event log files are located here: <site folder>/system/storage/logs/

## **prestashop compatibility**

The instructions for installing and configuring the plugin for PrestaShop version 1.7.x are given as example.

## **netcat compatibility**

The plugin is compatible with Netcat versions 5.9.x and higher and has been tested on E-commerce editions.

## **avoid error in personal account when testing REST**

This situation can be avoided in one of the following ways:

- use the Incognito mode for REST requests, and work in your personal account in the browser window as usual;
- use different browsers to test REST requests and work in your personal account.

## **wSDL connection URLs**

Use the following URL to connect to the Test Service (WSDL):  
<https://3dsec.sberbank.ru/payment/webservices/merchant-ws?wsdl>.



## payment reversal conditions

- With one-phase payments, payment reversal is possible for orders in the state **Deposited**, but only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.
- In case of two-phase payments, a payment reversal can be processed for orders in the **Approved** status.

## refund conditions

Refunds are possible for payments in **Deposited** status.

## passing clientId in payment by binding

When `clientId` parameter is passed initially, the client's identifier is registered in the payment gateway and the data of their card is saved (except for the CVC code). On subsequent passes of this parameter, the client is redirected to a page where they can choose from the following options:

- use the previously saved data of the payment card – in this case, the client does not need to re-enter the data of the payment card for payment;
- enter the data of the new payment card – in this case, the data of the new payment card will be additionally linked to the identifier of this client.

## what is a callback

Callback notifications are not email notifications or telephone notifications. These are notifications received through the programming interface.

## contact support to enable callback

To enable the ability to receive checksum notifications and get a private key, contact technical support.

## self-signed certificates cannot be used

Self-signed certificates are not allowed. The certificate must be signed by a trusted certification authority (see above).

## basic callback parameters

The table shows the basic parameters. A number of additional parameters that will be passed in notifications can be specified in personal account.

## **parameters must be sorted alphabetically**

Notably, the «parameter\_name;parameter\_value» pairs must be sorted in direct alphabetical order by parameter name.

## **checksum check value**

If the checksums match, this notification is authentic and was sent by the payment gateway. Otherwise, it is likely that the attacker is trying to pass off their notification as a payment gateway notification.

## **use Latin characters**

Use Latin characters when filling out the form.

## **need an apple developer account**

To be able to complete this procedure, you need to have an Apple Developer account.

## **apple pay id creation rules**

The identifier should begin with the word **merchant**. The address of your live site should be specified in the reverse order. For example, for the site **sale.test.ru** the identifier will look as **merchant.ru.test.sale**.

## **checklist for testing apple pay**

Below there is a short list of actions that will allow you to test Apple Pay. Further details are available [in documentation](#) on the Apple website.

## **how to create a test account with apple**

For more information about creating a test account, see [Apple documentation](#).

## **create a new apple account**

If you mistakenly use a test account on a test device to log in to a production environment (such as iTunes) instead of logging in to a test environment, this test account will become invalid and cannot be used again. In this case create a new test account with a new email address.

## **samsung only for payment via mobile app**

Steps 1–3 are necessary to enable payment via the mobile app.

## **it is not necessary to wait for the completion of the report**

You can start generating a new report without waiting for the completion of the previous one.

## **number of generated reports**

Number in brackets after the caption **Reports** means how many reports are currently being generated.

## **fiscalization settings are set by support**

Fiscalization settings are set by technical support upon request to [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru). At this stage, you do not need to configure the fiscalization parameters yourself.

## **information about actions in atol personal account**

Detailed information about actions in the ATOL personal account is given on [ATOL website](#) (external link).

## **test data for connection to OFD**

Do not specify actual data to connect to the fiscal data operator in the test personal account! If you do so, test payments will be registered with the fiscal data operator as if they were real. After entering the data into the live personal account – check that they are valid and saved correctly.

## **passing empty compositecompletionamount**

If zero is passed in this parameter, completion will occur for the entire pre-authorized amount.

## **required parameters in getorderstatusextended**

Either `orderId` or `orderNumber` must be present in the request. If both parameters are present in the request, `orderId` has priority.

## **required parameters in getOrderstatusextended ws**

The request must contain either orderId or merchantordernumber. If both parameters are present in the request, orderId has priority.

## **post only**

Only POST is supported.

## **the response depends on the OFD**

Some parameters to be returned may depend on the Fiscal Data Operator used.

## **use standard requests**

Use the standard requests to the payment gateway for the operations of reversal, refund and payment completion.

## **you need to pass the sbrf\_spasibo parameter**

If the merchant is connected to the «Spasibo from Sberbank» affiliate program, and bonus points are sent in the registration request, then the parameter should be passed as an additional parameter sbrf\_spasibo:amount\_bonus indicating the amount of bonus points to be deducted, as in the example below.

```
jsonParams={«sbrf_spasibo:amount_bonus»:«100»}
```

In this case, the amount in money will be deducted from the amount parameter.

## **you need to pass the sbrf\_spasibo parameter (ws)**

If the merchant is connected to the «Spasibo from Sberbank» affiliate program, and bonus points are sent in the registration request, then the parameter should be passed as an additional parameter sbrf\_spasibo:amount\_bonus indicating the amount of bonus points to be deducted, as in the example below.

```
<params name=«sbrf_spasibo:amount_bonus» value=«1030»/>
```

In this case, the amount in money will be deducted from the amount parameter.

## **reversals and refunds cannot be made**

Orders that initiate recurring payments cannot be reversed and refunded, as in these cases there is

no real debiting.

### **passing an empty amount**

If you do not specify the amount parameter, completion will take place for the entire pre-authorized amount.

### **quantity and positionid are sufficient**

For error-free processing of the request it is sufficient to pass the quantity and positionId parameters.

### **passing empty depositamount**

If you do not specify the depositAmount parameter, completion will take place for the entire pre-authorized amount.

### **authcode field is obsolete**

The order status must be determined by the orderStatus parameter value. The authCode field is obsolete.

### **getreceiptstatus is not supported in ws**

Currently, requesting information about the status of a check in the WS interface is not supported.

### **download virtuemart**

From the VirtueMart developer site you can [download](#) the Joomla! distribution with pre-installed add-on VirtueMart. The names of such combined archives end with **Stable-Full\_Package.zip**. When doing this, choose a combination of Joomla! and VirtueMart, compatible with the payment plugin.

### **joomla compatibility**

The plugins are only guaranteed to work with the following combinations of Joomla! and VirtueMart add-ons. If you use other combinations of versions (for example, after updating Joomla!) The plugin may not work correctly.

### **compliance with 54-fz**

Starting from this version, the plugin allows you to transfer a shopping cart, which provides the possibility of detailed fiscalization in accordance with [54-FZ](#).

### **on the example of the next version of joomla**

Installing and configuring the plugin is presented on Joomla! version 3.6.5 and VirtueMart version 3.2.0.

### **android pay not supported**

The Android Pay payment method is currently not supported by Google. This payment method will be replaced by Pay with Google (it is currently temporarily unavailable for integration, documentation is under development).

### **section in development**

The section is currently under construction.

### **itemamount is rounded to the second decimal place**

When calculating  $\text{itemAmount} = \text{itemPrice} * \text{quantity}$  parameter, the result is rounded up to the second digit after the decimal point. For example, if the calculation result is 100.255, then final value will be 100.26.

### **orderid ordernumber or merchantlogin required**

To execute the request, one of the following parameters must be specified:

- 2;
- 2;
- 2.

### **access to payment method**

To be able to accept payment in this way, you must have the respective rights in the system. If in doubt, contact technical support.

### **only one-phase payments**

Payment in this way is possible only when using a one-phase payment scheme.

## **asymmetric cryptography is recommended**

For greater security, use of a method in which the checksum is generated using asymmetric cryptography is recommended.

## **the order of the parameters can be arbitrary**

The order of parameters in the notification can be undefined.

## **symmetric cryptography example**

Below are examples of creating a checksum using symmetric cryptography.

## **in order to familiarize**

The information on this portal is provided as an introduction to the functionality of the payment gateway. The necessary data for connecting to the live environment will be provided after testing in accordance with [connection checklist](#).

## **request only through rest**

Currently this request is available through REST interface only.

## **for samsung pay web you need a javascript script**

To be able to receive payment via Samsung Pay Web with a merchant-side payment page, you must install [JavaScript script](#). When receiving a response from the payment gateway, you need to call the connect function of this script and pass in it certain parameters of the payment gateway response. For details, see the description of the corresponding [interaction schemes](#) and description of the [request](#).

## **https and tls are required**

Sites must use the HTTPS scheme and support the TLS 1.2 protocol.

## **safari required**

When making a payment from a web page, the buyer must be in the Safari browser.

## **programming skills required**

Programming skills are required to implement automatic check of payment completion.

## **don't share the private key with anyone**

Do not share or publish the private key to anyone. Signature verification must be implemented on the store's server. JavaScript implementation of signature verification on the front-end of the store is not allowed.

## **you need to have google merchant id**

To make payments from the website, your online store site must be registered with Google Pay.

## **for ofd, apply for connection**

Before registering a personal account of a cloud cash register provider [apply](#) to connect to Internet acquiring.

## **additionalparameters**

If the merchant has fiscalization set up, when indicating email (the buyer's email address) and/or phone (the buyer's cell phone number) as additional parameters, these parameters are first used to send the fiscal receipt.

## **test and live mode for plugins**

- If the plugin is running in test mode, in the respective fields (**Login** and **Password**) you should specify the data of the test service account.
- If the plugin works in the live mode, in the respective fields (**Login** and **Password**) you should specify the data of the live service account.

## **only if fiscalization is enabled**

This setting is applied only if you have configured fiscalization parameters – see details in [section on compliance with law 54-FZ](#).

## **fiscalization support for amiro**

Fiscalization is supported in plugins for Amiro.CMS versions 7.x and later (see [details on fiscalization](#)).



## **only for amiro 7 and higher**

The setting is available only in plugins for Amiro.CMS versions 7.x and later.

## **hold ctrl to select multiple items**

If you want to select multiple items, hold down the CTRL key on your keyboard before checking the items in the list.

## **plugin supports fiscalization**

The plugin supports fiscalization in accordance with the law [54-FZ](#).

## **features verify**

- Even if the payment amount is to be passed in the request, it will not be debited from the account.
- After the order has been successfully registered, it is put to the REVERSED (cancelled) status.

## **integration is not needed under the following conditions**

Integration with a payment gateway is not required if you use:

- [e-mail invoicing service](#);
- [payment links generator](#);
- work using [Yoom.Shop](#) (external ink);
- work using the following CMS:
  - [1C-Bitrix](#);
  - [WordPress](#);
  - [Joomla!](#);
- use the transfer of goods by default from the fiscalization settings of the Sberbank personal account.

## **checking fiscalization on the live server**

After making all the settings, we ask you to carefully check the correct filling of all fields in relation to the fiscalization parameters in the personal account of the bank, because the correctness of the transfer of fiscal data to the OFD and tax depends on this. We also recommend checking «in action» when sending a fiscal receipt.

In case of any difficulties or questions, you can send them to the address of the technical support of the bank [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru) and Business.Ru [support@mail.business.ru](mailto:support@mail.business.ru).

## google pay developer info

Information for developers is available in [Google documentation](#).

### information for Google Pay Web developers

- Information for developers is available in [Google documentation](#).
- The payment page must be designed in accordance with [brand guidelines](#).

### information for Google Pay In-App developers

- Information for developers is available in [Google documentation](#).
- The application must be designed in accordance with [brand guidelines](#).

## google tos

By accepting payments via Google Pay, you agree to [Google terms of service](#).

### passing the shopping cart is optional

If the cart and fiscalization data are not included in the request, then the default values specified in the personal account are passed to the Fiscal Data Operator (see User Manual for personal account).

### rest request type

Using the REST protocol, you should send requests with the type `application/x-www-form-urlencoded`, rather than `multipart/form-data`.

### not enough rights to submit a payment form

If the section **Invoice to email** the following message is displayed: “You do not have sufficient rights to send the payment form”, please contact the technical support service [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru) with a request to activate the invoicing option. Make sure to include the following information in your request:

- operator login (used to enter your personal account).
- environment (test/production) for which changes need to be made

### QR code for payment link


The link can also be encoded as a QR code.

## **closing Merchant Portal session**

If there are no user actions for **10+ minutes**, the session is automatically closed.

## **exporting charts from Merchant Portal**



A click on  saves the chart image.

## **you can select several statuses**

You can set several statuses at the same time by pressing CTRL and selecting them with a left-click.

## **short payment term**

If you set the invoice validity period too short, the client will not be able to use the same invoice to pay for the order after some time.

## **use of two-phase payment**

When using a two-phase payment, it is recommended to debit the amount no later than 7 calendar days (for some industries, such as the hotel business, car rental, etc., the period of possible holding may be increased).

## **change password for production**

Passwords provided for both logins in production environment must be changed in the personal account (available [here](#)).

## **make sure to change password for all users**

Password for each login must be changed.

## **two-phase payment**

When using **two-phase** payment, it is recommended to debit the amount no later than 7 calendar days (for some industries, such as the hotel business, car rental, etc., the period of possible holding may be increased).

## **plugins support 54-fz**

### **Fiscalization - 54-FZ**

Most of the plugins presented support fiscalization. To clarify whether the plugin you are interested in supports the fiscalization functionality, you can visit our technical support address – [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru).

### **support for plugins**

We are ready to develop plugin support for new CMS. To do this, you need to contact the technical support service – [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru).

### **payment phases in Online Lending**

For online lending, only one-phase payment is applied.

#### **only one-phase payment**

When using the Online Lending functionality, only the one-phase payment is available.

#### **token authentication not possible**

When using the Online Lending functionality, token authentication is not possible. Login and password should be used for authentication.

#### **only through rest**

Online Lending functionality is available through the REST interface only.

#### **itemamount gift**

If all the items in the request are delivered without payment (a gift to customer), the value 0 should be passed.

#### **itemprice gift**

If the item is delivered without payment (a gift to customer), the value 0 should be passed.

## **can be changed in refund**

When requesting a refund, the value of this parameter may differ from that given during the registration of the order.

## **passing additionalOfdParams**

The transfer of this block is possible only when using the following OFDs:

- ATOL;
- Business.Ru;
- Evotor.

## **ffd 105 and higher**

The settings are effective only if the format of fiscal documents of version 1.05 and higher is selected.

## **FFD version must match**

The format of the version must match the format selected in the personal account of the bank and in the account of the fiscalization service.

## **add json header**

It is necessary to add to the request a header with the definition of the type of contents – «Content-Type: application/json» to ensure that the request is processed correctly.

## **assumes joomshopping is installed**

This tutorial assumes that your Joomla! the JoomShopping component is already installed.

## **assumes virtuemart is installed**

This tutorial assumes that your Joomla! the VirtueMart component is already installed.

## **simpla version**

The plugin was tested for Simpla version 3.2.x.

## google pay registration

These steps are necessary if you want to receive encrypted payment actions directly from Google Pay and redirect them to a payment gateway. Namely:

- online payment, if the payment page is located on the merchant's side;
- payment from the application.

## google pay info

For more information see [Integration Checklist](#) (external link).

## google pay create account

It is recommended to use an existing email address on your domain, for this you need to select the option **Use current email address**.

## samsung pay registration

These actions are necessary to enable payment via the mobile app. You do not need to take these steps to pay through the site.

## shop-script version

Shop-Script version must be 7.x or higher.

## not for ofd.ru

Not applicable for **OFD.ru**.

## magento compatibility

Magento plugins for versions 2.2.x and 2.3.x are incompatible. When upgrading Magento version from 2.2.x to 2.3.x, you must first uninstall the plugin for version 2.2.x and then install the plugin for version 2.3.x.

## php folder

The beginning of the command depends on the folder where PHP is installed. In particular, the PHP version can be specified without a hyphen: php5.6, php7.1, php7.2, etc.

If in doubt, contact your hosting provider.

### **may not have enough memory**

On shared hosting, the last command may not be executed due to the memory limit set by the hosting provider. In this case, contact your hosting provider.

### **uninstalling magento**

In code folder, as well as in the folders of a higher level, (app) there may be data not related to the plugin, so delete only the Rbs folder and its contents.

### **6 and 8 are not used**

Types 6 (gambling winnings) and 8 (lottery winnings) are not used.

### **initpro start of integration**

At this step, you can right away configure [automatic addition of your cash registers](#) to the personal account of the OFD.

### **initpro requires an EDS**

To register a cash desk through the FTS website, you will need an EDS. If you don't have it, you can [order it from us](#) or submit a paper application for registering the cash register with the tax service.

### **initpro free code**

You can find the free OFD activation code in [cash register profile](#) in the personal account of Initpro.

### **no integration setup required**

You don't need integration setup if:

- you are using [email invoicing service](#);
- you are using [payment link generator](#);
- you are working with the following CMSs: [1C-Bitrix](#), [WordPress](#), [Joomla!](#);
- use the default product transfer from [fiscalization settings](#) of Merchant Portal

Sberbank.

## **closeOfdReceipt**

- A closing receipt can be sent multiple times for one order. There is no limit on the amount, that is, the amount indicated in the closing receipt may be less or greater than the order amount.
- An exception is made for orders for which a refund has been made. For orders for which a refund was made, the closing of the check can only be done if the total of all refunds is less than the confirmed order amount (i.e., there was no full refund).

## **closing if refund**

Closing a check can only be done if a partial, not full refund has been made for the order.

## **amount info**

If you specify amount=0 in the request, a refund is made for the entire amount of the order.

## **amount credit**

Before summing up all line items, the product of quantity and price for each good item is rounded to an integer. If the value is followed by 5 or greater after the decimal point, it is rounded up. Below there are examples of rounding.

1. If quantity = 0.111, and price = 5500, then the result is 611 (610.5 rounded up).
2. If quantity = 1.455, and price = 6900, then the result is 10040 (10039.5 rounded up).
3. If quantity = 1.211, and price = 6988, then the result is 8462 (8462.468 rounded down).

Therefore, the «amount» parameter value must be equal to the sum of the line items rounded.

## **credit only rest**

Online Lending functionality is available through the REST interface only.

## **items credit warning**

Do not use «'» character combination inside this section, or it will cause an error on the gateway side.

## **itemprice info**

- itemPrice cannot be less than 0.
- itemPrice may be either string or integer.



## **getreceiptstatus warning**

To execute the request, the order number or its identifier in the fiscalizator must be specified.

## **customerdetails info credit**

A parameter included in a block and indicated as mandatory is only mandatory within that block. If the block itself is not mandatory and is not passed, its included parameters should not be passed.

## **the ability to make two-phase payments**

Two-phase payments functionality is available if specifically requested by Merchant from the bank.

## **preauth parameter**

This parameter can be passed either as a boolean value (without quotation marks) or as a string (with quotation marks).

## **email ОФД**

Most OFDs support delivery of receipt only to one email address. Some OFDs allow receipt delivery to multiple emails. In this case, the emails should be separated by commas without spaces. Contact your OFD for availability of receipt delivery to multiple emails.

## **positionid warning**

The registered cart must contain an item with this positionId value, otherwise the order will fail.

## **callback delimiter**

Use semicolon («;») without a space as the separator. The string should end with semicolon («;») as well.

## **sending a notification is considered unsuccessful**

If a response other than the 200 OK HTTP code is returned to the payment gateway, the notification sending is considered to be unsuccessful. In this case, the payment gateway repeats sending the notification (the first attempt is made after 30 seconds, and the next with an interval of 10 minutes) until one of the following conditions is met:

- the payment gateway receives the 200 OK HTTP code in response to the callback notification

or

- there are six unsuccessful attempts to inform in sequence.

When either of the above conditions is met, attempts to send a callback notification about the operation are terminated.

## **REST authorization**

The transfer of authorization data (username/password) can be transferred both in the request body and in the header.

## **standard refund and reversal requests**

Refund and reversal requests are standard payment gateway requests.

## **token from tech support**

The token value can be obtained from the technical support staff.

## **autocomplete date**

To enable this functionality, please contact technical support.

## **item details warning**

The field size limit is 1024 bytes.

## **name (items)**

Use

- to pass \ use \«- to pass »

## **phone note**

If the number is passed in a separate parameter and in additional parameters, the number specified in this phone parameter will be used.

## **customization of the payment page**

Currently, minimal customization of the payment page is available, only the following elements can be changed:

- logo;
- footer.

The requirements for these elements are presented in the instructions for working with your personal account.

## **recurrent payment auto payments**

For recurring payments via SberPay, only one-phase payment can be used. At the same time, for ordinary bindings, a two-phase payment is possible.

## **data types**

In this document, the following data type conventions are used when describing request and response parameters:

- A<n> - a sequence of Latin letters of length <n>;
- A..<n> - a sequence of Latin letters with a length not exceeding <n>;
- N<n> - a sequence of digits of length <n>;
- N..<n> - a sequence of digits with a length not exceeding <n>;
- AN<n> - a sequence of Latin letters and numbers of fixed length <n>;
- AN.. <n> - a sequence of Latin letters and numbers with a length not exceeding <n>;
- ANS<n> - a sequence of Latin letters, numbers and characters of fixed length <n>
- ANS.. <n> - a sequence of Latin letters, numbers and characters with a length not exceeding <n>;
- UTC - date and time, in this case: the date must be passed without specifying the time zone, Moscow time, for the SOAP protocol, the standard encoding xs: dateTime is used.

## **data types 2**

In this document, the following data type conventions are used when describing request and response parameters:

- string - string;
- array - data array;
- object - data object;
- data - date;
- boolean - boolean (logical) data type;
- int. (integer) - integer data type;
- amount - a numeric data type (amount).

## **ATOL phone**

When using ATOL fiscalization, the phone parameter should be sent in the following format:

The phone number must be passed together with the country code without spaces and additional characters except for the quotation marks (the number «371 2 1234567» must be passed as «+37121234567»). If the phone number belongs to Russia (prefix «+7»), then the value can be passed without a prefix (the number «+79251234567» can be sent as «9251234567»). The maximum string length is 64 characters. One of the fields must be filled in the request: email or phone.

### **reserved names in params, jsonParams, additionalParameters (Spasibo)**

It is forbidden to pass reserved names in the parameter (if they are passed, the order may be rejected):

- sbrf\_spasibo:amount\_bonus
- sbrf\_sbermiles:amount\_bonus
- loyaltyId

### **possible values of measure**

Possible values of **measure**

Value	Description
0	Used for payment subjects that can be sold individually or in units (as well as in the event that the payment subject is a product subject to mandatory labeling with an identification tool (which can be known from mark_code being passed))
10	Gram
11	Kilogram
12	Ton
20	Centimeter
21	Decimeter
22	Meter
30	Square centimeter
31	Square decimeter
32	Square meter
40	Milliliter
41	Liter
42	Cubic meter
50	Kilowatt hour
51	Gigacalorie
70	Day (24h)
71	Hour
72	Minute
73	Second
80	Kilobyte
81	Megabyte
82	Gigabyte
83	Terabyte
255	Applicable when using other units of measurement

## markQuantity

Parameters of **markQuantity** tag:

*A parameter included in a tag and indicated as mandatory is only mandatory within that tag. If the tag itself is not mandatory and is not passed, its included parameters should not be passed.*

Name	Type	Mandatory	Description
numerator	N..12	yes	The numerator of the fractional part of the payment subject.
denominator	N..12	yes	The denominator of the fractional part of the payment subject.

The table below shows how the merchant and the payment gateway interact when a payment is refunded.

1	The merchant makes a request for refund to the buyer: <ul style="list-style-type: none"><li>• <a href="#">REST interface (refund.do)</a>;</li><li>• <a href="#">WSDL interface (refundOrder)</a>.</li></ul>
2	After receiving a successful response from the Payment Gateway, the Merchant makes a request for order status: <ul style="list-style-type: none"><li>• <a href="#">REST interface (getOrderStatusExtended.do)</a>;</li><li>• <a href="#">WSDL interface (getOrderStatusExtended)</a>.</li></ul>

There is an option to set up [callback notifications](#) about order status instead of sending the getOrderStatusExtended request. These notifications will be sent to you automatically after the order status changes.

## Payment with registration of receipts on POS equipment

### Standard API

#### Payment with passing of shopping cart and data for registering a check

	One-phase payment	Two-phase payment
1	The payer chooses goods or services on the merchant website and picks bank card as a payment option.	

One-phase payment	Two-phase payment
<p>The merchant sends an order registration request to the payment gateway:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (register.do)</a>;</li> <li>• <a href="#">WSDL interface (registerOrder)</a>.</li> </ul> <p>Among others, the following parameters are passed in the request:</p> <ul style="list-style-type: none"> <li>• returnUrl – the URL to which the payer will be redirected if payment is successful;</li> <li>• failUrl – the URL to which the payer will be redirected if payment is unsuccessful.</li> <li>• orderBundle – a block containing a cart of goods and data for registering a receipt;</li> <li>• taxSystem – tax system identifier.</li> </ul>	

## Payment reversal

Reversal of a payment for an order is available to Merchants provided that they have the corresponding permissions and approval of Sberbank.

- With one-phase payments, payment reversal is possible for orders in the state **Deposited**, but only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.
- In case of two-phase payments, a payment reversal can be processed for orders in the **Approved** status.

The table below shows how the merchant and the payment gateway interact when a payment is reversed.

<p>The merchant makes a request to reverse the payment:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (reverse.do)</a>;</li> <li>• <a href="#">WSDL interface (reverseOrder)</a>.</li> </ul>	<p>After receiving a successful response from the Payment Gateway, the Merchant makes a request for order status:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do)</a>;</li> <li>• <a href="#">WSDL interface (getOrderStatusExtended)</a>.</li> </ul>
--	---

There is an option to set up [callback notifications](#) about order status instead of sending the getOrderStatusExtended request. These notifications will be sent to you automatically after the order status changes.

## Refunds to the payer

The Payment Gateway allows full or partial refunds for paid orders.

### **callback instead of getOrderStatusExtended**

There is an option to set up [callback notifications](#) about order status instead of sending the getOrderStatusExtended request. These notifications will be sent to you automatically after the order status changes.

### **getreceiptstatus**

To get information about a sales receipt (for example, to then pass this data to a customer) use the [getReceiptStatus.do](#) request. Currently, this request can only be made through the REST interface.

### **recommended wordpress version**

Recommended version of WordPress for working with the Sberbank payment plugin: not lower than 4.8.3.

### **download woocommerce**

You can download the required version of WooCommerce on [github.com](#) (link leads to WooCommerce version history).

### **woocommerce compatibility**

The WooCommerce plugin is not compatible with all WordPress themes. Here, the [Storefront](#) theme is used as example. This theme is designed by WooCommerce developers.

### **meanings of terms**

Meanings of terms highlighted on this page in **bold** font are given in the [glossary](#).

### **types of payments**

One-phase payments are preferable if the product or service is provided immediately after payment. Two-phase payments should be used if some time elapses between the buyer's decision to pay and the delivery of the selected product or service.

### **need to check 3dsec**

The need for verification using 3-D Secure depends on the buyer's bank card, as well as on the settings configured for the Merchant upon agreement with Sberbank.

### **reversal before midnight**

In case of one-phase payments, the reversal operation is possible only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.

### **access to personal account**

Regardless of the chosen connection method, you are given access to the [personal account](#) on the payment gateway server.

### **you can use test cards**

You can use [test cards](#) to try out the functionality.

### **access to payment methods**

The available payment acceptance methods depend on settings configured for you. If in doubt, check with a Sberbank representative which method of accepting payments you can use.

### **passed as clientid**

The value of this parameter goes in the registration request as the `clientId` parameter.

### **need to use utf-8**

The page of your site on which the payment solution will be used must be saved in UTF-8 encoding, and it must also have a meta-tag `<meta name=«viewport» content=«width=device-width»>`.

### **update the links**

The links point to test environment. Once done with testing, update the links so that they point to the production environment.



**tls is required**

The server on which the content management system is located must support the TLS 1.2 protocol. If in doubt, check with your hosting provider.

**minimum amount to be debited**

The minimum amount of funds to be debited is 100 rubles.

**default parameters**

If the cart and fiscalization data are not passed in the request, the default values specified in the personal account are passed to the Fiscal Data Operator (see [setting up fiscalization data](#) for more details).

**passing a field to processing**

To enable sending this field to the processing system, contact the technical support service.

**parameter must be specified**

Specifying this parameter when processing payments with the use of bindings is mandatory. Otherwise, a payment will be unsuccessful.

**fields are passed by default**

By default, the following fields are passed to the Bank processing system:

- `orderNumber` – order number in the store system;
- `description` – order description (no more than 24 characters; %, +, carriage return \r, and line feed \n cannot be used).

**mandatory to pass either email or phone**

It is mandatory to pass one of the two parameters: email or phone.

**parameters are not required if the block is missing**

A parameter included in a block and indicated as mandatory is only mandatory within that block. If the section is optional and missing, then the parameters included in it should not be passed.

### **access to personal account**

Access to personal account is granted regardless of the chosen connection method.

### **access to functions depends on settings**

Functions available in the personal account depend on the configured settings. If in doubt, please contact a Sberbank representative.

### **more details available on the 1c-bitrix website**

For more details on the differences between editions, see the website «[1C-Bitrix](#)».

### **1c-bitrix version used as example**

1C-Bitrix version used for installation and configuration: **1C-Bitrix: Site Manager, Small Business Edition**, version **16.5.4**.

### **terms in the 1c-bitrix interface**

In the 1C-Bitrix interface, a «payment system» is a payment method that a customer can choose, for example: «in cash to a courier», «by bank card», etc. The term «payment method» is used throughout this description.

### **1C-Bitrix settings order**

The order of fields and settings may differ depending on the edition of your 1C-Bitrix instance.

### **1C-Bitrix using default tab as example**

In this description, the setting is shown using the example of the **Default** tab. If the tabs **Individual** and **Entity** have not been edited before, the changes made will apply to these tabs.

### **password if the plugin is in test mode**

If the plugin is running in test mode, you should specify the password for the test user account. If the plugin is running in standard (production) mode, you should specify the password for the production user account.

### **login if the plugin is in test mode**

If the plugin is running in test mode, you should specify the login for the test user account. If the plugin is running in production mode, you should specify the password for the production user account.

### **wordpress display order**

If you want to change the order in which a payment method is displayed for a customer, click on the icon opposite the required payment method and, while holding the left mouse button, move this payment method to the required position.

### **cs-cart compatibility**

The payment plugin was tested for compatibility with CS-Cart version 4.6.x.

### **terms in the cs-cart interface**

In the CS-Cart interface, modules that extend the site's functionality are called «add-ons». In this description, the term «plugin» is used.

### **plugin settings only**

The description contains only those settings that relate directly to the payment plugin. The rest of the settings are related to the content management system (CMS).

### **terms in the Drupal interface**

In the Drupal interface, add-ons are called «modules». The payment plugin falls into the same category, however the term «plugin» is used in this description.

### **drupal description based on example**

A combined distribution [Commerce Kickstart](#) version 7.x-2.45 was used for the purposes of this description. This distribution combines Drupal version 7.54 and Drupal Commerce version 7.x-1.13.

### **opencart compatibility**

The plugin was tested on OpenCart/ocStore version 2.3.0.2.

### the location of the log files

The event log files are located here: <site folder>/system/storage/logs/

### prestashop compatibility

The instructions for installing and configuring the plugin for PrestaShop version 1.7.x are given as example.

### netcat compatibility

The plugin is compatible with Netcat versions 5.9.x and higher and has been tested on E-commerce editions.

### avoid error in personal account when testing REST

This situation can be avoided in one of the following ways:

- use the Incognito mode for REST requests, and work in your personal account in the browser window as usual;
- use different browsers to test REST requests and work in your personal account.

### wSDL connection URLs

Use the following URL to connect to the Test Service (WSDL):  
<https://3dsec.sberbank.ru/payment/webservices/merchant-ws?wsdl>.

### payment reversal conditions

- With one-phase payments, payment reversal is possible for orders in the state **Deposited**, but only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.
- In case of two-phase payments, a payment reversal can be processed for orders in the **Approved** status.

### refund conditions

Refunds are possible for payments in **Deposited** status.

### passing clientId in payment by binding

When `clientId` parameter is passed initially, the client's identifier is registered in the payment gateway and the data of their card is saved (except for the CVC code). On subsequent passes of this parameter, the client is redirected to a page where they can choose from the following options:

- use the previously saved data of the payment card – in this case, the client does not need to re-enter the data of the payment card for payment;
- enter the data of the new payment card – in this case, the data of the new payment card will be additionally linked to the identifier of this client.

#### **what is a callback**

Callback notifications are not email notifications or telephone notifications. These are notifications received through the programming interface.

#### **contact support to enable callback**

To enable the ability to receive checksum notifications and get a private key, contact technical support.

#### **self-signed certificates cannot be used**

Self-signed certificates are not allowed. The certificate must be signed by a trusted certification authority (see above).

#### **basic callback parameters**

The table shows the basic parameters. A number of additional parameters that will be passed in notifications can be specified in personal account.

#### **parameters must be sorted alphabetically**

Notably, the «parameter\_name;parameter\_value» pairs must be sorted in direct alphabetical order by parameter name.

#### **checksum check value**

If the checksums match, this notification is authentic and was sent by the payment gateway. Otherwise, it is likely that the attacker is trying to pass off their notification as a payment gateway notification.

#### **use Latin characters**

Use Latin characters when filling out the form.

#### **need an apple developer account**

To be able to complete this procedure, you need to have an Apple Developer account.

#### **apple pay id creation rules**

The identifier should begin with the word **merchant**. The address of your live site should be specified in the reverse order. For example, for the site **sale.test.ru** the identifier will look as **merchant.ru.test.sale**.

#### **checklist for testing apple pay**

Below there is a short list of actions that will allow you to test Apple Pay. Further details are available [in documentation](#) on the Apple website.

#### **how to create a test account with apple**

For more information about creating a test account, see [Apple documentation](#).

#### **create a new apple account**

If you mistakenly use a test account on a test device to log in to a production environment (such as iTunes) instead of logging in to a test environment, this test account will become invalid and cannot be used again. In this case create a new test account with a new email address.

#### **samsung only for payment via mobile app**

Steps 1–3 are necessary to enable payment via the mobile app.

#### **it is not necessary to wait for the completion of the report**

You can start generating a new report without waiting for the completion of the previous one.

#### **number of generated reports**

Number in brackets after the caption **Reports** means how many reports are currently being generated.

### **fiscalization settings are set by support**

Fiscalization settings are set by technical support upon request to [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru). At this stage, you do not need to configure the fiscalization parameters yourself.

### **information about actions in atol personal account**

Detailed information about actions in the ATOL personal account is given on [ATOL website](#) (external link).

### **test data for connection to OFD**

Do not specify actual data to connect to the fiscal data operator in the test personal account! If you do so, test payments will be registered with the fiscal data operator as if they were real. After entering the data into the live personal account – check that they are valid and saved correctly.

### **passing empty compositecompletionamount**

If zero is passed in this parameter, completion will occur for the entire pre-authorized amount.

### **required parameters in getorderstatusextended**

Either `orderId` or `orderNumber` must be present in the request. If both parameters are present in the request, `orderId` has priority.

### **required parameters in getorderstatusextended ws**

The request must contain either `orderId` or `merchantordernumber`. If both parameters are present in the request, `orderId` has priority.

### **post only**

Only POST is supported.

### **the response depends on the OFD**

Some parameters to be returned may depend on the Fiscal Data Operator used.

### **use standard requests**

Use the standard requests to the payment gateway for the operations of reversal, refund and payment completion.

### **you need to pass the sbrf\_spasibo parameter**

If the merchant is connected to the «Spasibo from Sberbank» affiliate program, and bonus points are sent in the registration request, then the parameter should be passed as an additional parameter `sbrf_spasibo:amount_bonus` indicating the amount of bonus points to be deducted, as in the example below.

```
jsonParams={«sbrf_spasibo:amount_bonus»:«100»}
```

In this case, the amount in money will be deducted from the amount parameter.

### **you need to pass the sbrf\_spasibo parameter (ws)**

If the merchant is connected to the «Spasibo from Sberbank» affiliate program, and bonus points are sent in the registration request, then the parameter should be passed as an additional parameter `sbrf_spasibo:amount_bonus` indicating the amount of bonus points to be deducted, as in the example below.

```
<params name=«sbrf_spasibo:amount_bonus» value=«1030»/>
```

In this case, the amount in money will be deducted from the amount parameter.

### **reversals and refunds cannot be made**

Orders that initiate recurring payments cannot be reversed and refunded, as in these cases there is no real debiting.

### **passing an empty amount**

If you do not specify the amount parameter, completion will take place for the entire pre-authorized amount.

### **quantity and positionid are sufficient**

For error-free processing of the request it is sufficient to pass the quantity and positionId parameters.



### **passing empty depositamount**

If you do not specify the `depositAmount` parameter, completion will take place for the entire pre-authorized amount.

### **authcode field is obsolete**

The order status must be determined by the `orderStatus` parameter value. The `authCode` field is obsolete.

### **getreceiptstatus is not supported in ws**

Currently, requesting information about the status of a check in the WS interface is not supported.

### **download virtuemart**

From the VirtueMart developer site you can [download](#) the Joomla! distribution with pre-installed add-on VirtueMart. The names of such combined archives end with **Stable-Full\_Package.zip**. When doing this, choose a combination of Joomla! and VirtueMart, compatible with the payment plugin.

### **joomla compatibility**

The plugins are only guaranteed to work with the following combinations of Joomla! and VirtueMart add-ons. If you use other combinations of versions (for example, after updating Joomla!) The plugin may not work correctly.

### **compliance with 54-fz**

Starting from this version, the plugin allows you to transfer a shopping cart, which provides the possibility of detailed fiscalization in accordance with [54-FZ](#).

### **on the example of the next version of joomla**

Installing and configuring the plugin is presented on Joomla! version 3.6.5 and VirtueMart version 3.2.0.

### **android pay not supported**

The Android Pay payment method is currently not supported by Google. This payment method will be replaced by Pay with Google (it is currently temporarily unavailable for integration, documentation is under development).

## **section in development**

The section is currently under construction.

## **itemamount is rounded to the second decimal place**

When calculating  $\text{itemAmount} = \text{itemPrice} * \text{quantity}$  parameter, the result is rounded up to the second digit after the decimal point. For example, if the calculation result is 100.255, then final value will be 100.26.

## **orderid ordernumber or merchantlogin required**

To execute the request, one of the following parameters must be specified:

- 2;
- 2;
- 2.

## **access to payment method**

To be able to accept payment in this way, you must have the respective rights in the system. If in doubt, contact technical support.

## **only one-phase payments**

Payment in this way is possible only when using a one-phase payment scheme.

## **asymmetric cryptography is recommended**

For greater security, use of a method in which the checksum is generated using asymmetric cryptography is recommended.

## **the order of the parameters can be arbitrary**

The order of parameters in the notification can be undefined.

## **symmetric cryptography example**

Below are examples of creating a checksum using symmetric cryptography.

## **in order to familiarize**

The information on this portal is provided as an introduction to the functionality of the payment gateway. The necessary data for connecting to the live environment will be provided after testing in accordance with [connection checklist](#).

## **request only through rest**

Currently this request is available through REST interface only.

## **for samsung pay web you need a javascript script**

To be able to receive payment via Samsung Pay Web with a merchant-side payment page, you must install [JavaScript script](#). When receiving a response from the payment gateway, you need to call the connect function of this script and pass in it certain parameters of the payment gateway response. For details, see the description of the corresponding [interaction schemes](#) and description of the [request](#).

## **https and tls are required**

Sites must use the HTTPS scheme and support the TLS 1.2 protocol.

## **safari required**

When making a payment from a web page, the buyer must be in the Safari browser.

## **programming skills required**

Programming skills are required to implement automatic check of payment completion.

## **don't share the private key with anyone**

Do not share or publish the private key to anyone. Signature verification must be implemented on the store's server. JavaScript implementation of signature verification on the front-end of the store is not allowed.

## **you need to have google merchant id**

To make payments from the website, your online store site must be registered with Google Pay.

### **for ofd, apply for connection**

Before registering a personal account of a cloud cash register provider [apply](#) to connect to Internet acquiring.

### **additionalparameters**

If the merchant has fiscalization set up, when indicating email (the buyer's email address) and/or phone (the buyer's cell phone number) as additional parameters, these parameters are first used to send the fiscal receipt.

### **test and live mode for plugins**

- If the plugin is running in test mode, in the respective fields (**Login** and **Password**) you should specify the data of the test service account.
- If the plugin works in the live mode, in the respective fields (**Login** and **Password**) you should specify the data of the live service account.

### **only if fiscalization is enabled**

This setting is applied only if you have configured fiscalization parameters – see details in [section on compliance with law 54-FZ](#).

### **fiscalization support for amiro**

Fiscalization is supported in plugins for Amiro.CMS versions 7.x and later (see [details on fiscalization](#)).

### **only for amiro 7 and higher**

The setting is available only in plugins for Amiro.CMS versions 7.x and later.

### **hold ctrl to select multiple items**

If you want to select multiple items, hold down the CTRL key on your keyboard before checking the items in the list.

### **plugin supports fiscalization**

The plugin supports fiscalization in accordance with the law [54-FZ](#).

## **features verify**

- Even if the payment amount is to be passed in the request, it will not be debited from the account.
- After the order has been successfully registered, it is put to the REVERSED (cancelled) status.

## **integration is not needed under the following conditions**

Integration with a payment gateway is not required if you use:

- [e-mail invoicing service](#);
- [payment links generator](#);
- work using [Yoom.Shop](#) (external ink);
- work using the following CMS:
  - [1C-Bitrix](#);
  - [WordPress](#);
  - [Joomla!](#);
- use the transfer of goods by default from the fiscalization settings of the Sberbank personal account.

## **checking fiscalization on the live server**

After making all the settings, we ask you to carefully check the correct filling of all fields in relation to the fiscalization parameters in the personal account of the bank, because the correctness of the transfer of fiscal data to the OFD and tax depends on this. We also recommend checking «in action» when sending a fiscal receipt.

In case of any difficulties or questions, you can send them to the address of the technical support of the bank [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru) and Business.Ru [support@mail.business.ru](mailto:support@mail.business.ru).

## **google pay developer info**

Information for developers is available in [Google documentation](#).

### **information for Google Pay Web developers**

- Information for developers is available in [Google documentation](#).
- The payment page must be designed in accordance with [brand guidelines](#).

### **information for Google Pay In-App developers**

- Information for developers is available in [Google documentation](#).
- The application must be designed in accordance with [brand guidelines](#).

## google tos

By accepting payments via Google Pay, you agree to [Google terms of service](#).

## passing the shopping cart is optional

If the cart and fiscalization data are not included in the request, then the default values specified in the personal account are passed to the Fiscal Data Operator (see User Manual for personal account).

## rest request type

Using the REST protocol, you should send requests with the type `application/x-www-form-urlencoded`, rather than `multipart/form-data`.

## not enough rights to submit a payment form

If the section **Invoice to email** the following message is displayed: “You do not have sufficient rights to send the payment form”, please contact the technical support service [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru) with a request to activate the invoicing option. Make sure to include the following information in your request:

- operator login (used to enter your personal account).
- environment (test/production) for which changes need to be made

## QR code for payment link

The link can also be encoded as a QR code.

## closing Merchant Portal session

If there are no user actions for **10+ minutes**, the session is automatically closed.

## exporting charts from Merchant Portal



A click on  saves the chart image.

## you can select several statuses

You can set several statuses at the same time by pressing CTRL and selecting them with a left-click.

### **short payment term**

If you set the invoice validity period too short, the client will not be able to use the same invoice to pay for the order after some time.

### **use of two-phase payment**

When using a two-phase payment, it is recommended to debit the amount no later than 7 calendar days (for some industries, such as the hotel business, car rental, etc., the period of possible holding may be increased).

### **change password for production**

Passwords provided for both logins in production environment must be changed in the personal account (available [here](#)).

### **make sure to change password for all users**

Password for each login must be changed.

### **two-phase payment**

When using **two-phase** payment, it is recommended to debit the amount no later than 7 calendar days (for some industries, such as the hotel business, car rental, etc., the period of possible holding may be increased).

### **plugins support 54-fz**

### **Fiscalization - 54-FZ**

Most of the plugins presented support fiscalization. To clarify whether the plugin you are interested in supports the fiscalization functionality, you can visit our technical support address – [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru).

### **support for plugins**

We are ready to develop plugin support for new CMS. To do this, you need to contact the technical support service – [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru).

### **payment phases in Online Lending**

For online lending, only one-phase payment is applied.

**only one-phase payment**

When using the Online Lending functionality, only the one-phase payment is available.

**token authentication not possible**

When using the Online Lending functionality, token authentication is not possible. Login and password should be used for authentication.

**only through rest**

Online Lending functionality is available through the REST interface only.

**itemamount gift**

If all the items in the request are delivered without payment (a gift to customer), the value 0 should be passed.

**itemprice gift**

If the item is delivered without payment (a gift to customer), the value 0 should be passed.

**can be changed in refund**

When requesting a refund, the value of this parameter may differ from that given during the registration of the order.

**passing additionalOfdParams**

The transfer of this block is possible only when using the following OFDs:

- ATOL;
- Business.Ru;
- Evotor.

**ffd 105 and higher**

The settings are effective only if the format of fiscal documents of version 1.05 and higher is selected.



## **FFD version must match**

The format of the version must match the format selected in the personal account of the bank and in the account of the fiscalization service.

## **add json header**

It is necessary to add to the request a header with the definition of the type of contents - «Content-Type: application/json» to ensure that the request is processed correctly.

## **assumes joomshopping is installed**

This tutorial assumes that your Joomla! the JoomShopping component is already installed.

## **assumes virtuemart is installed**

This tutorial assumes that your Joomla! the VirtueMart component is already installed.

## **simpla version**

The plugin was tested for Simpla version 3.2.x.

## **google pay registration**

These steps are necessary if you want to receive encrypted payment actions directly from Google Pay and redirect them to a payment gateway. Namely:

- online payment, if the payment page is located on the merchant's side;
- payment from the application.

## **google pay info**

For more information see [Integration Checklist](#) (external link).

## **google pay create account**

It is recommended to use an existing email address on your domain, for this you need to select the option **Use current email address**.

### **samsung pay registration**

These actions are necessary to enable payment via the mobile app. You do not need to take these steps to pay through the site.

### **shop-script version**

Shop-Script version must be 7.x or higher.

### **not for ofd.ru**

Not applicable for **OFD.ru**.

### **magento compatibility**

Magento plugins for versions 2.2.x and 2.3.x are incompatible. When upgrading Magento version from 2.2.x to 2.3.x, you must first uninstall the plugin for version 2.2.x and then install the plugin for version 2.3.x.

### **php folder**

The beginning of the command depends on the folder where PHP is installed. In particular, the PHP version can be specified without a hyphen: php5.6, php7.1, php7.2, etc.

If in doubt, contact your hosting provider.

### **may not have enough memory**

On shared hosting, the last command may not be executed due to the memory limit set by the hosting provider. In this case, contact your hosting provider.

### **uninstalling magento**

In code folder, as well as in the folders of a higher level, (app) there may be data not related to the plugin, so delete only the Rbs folder and its contents.

### **6 and 8 are not used**

Types 6 (gambling winnings) and 8 (lottery winnings) are not used.

## **initpro start of integration**

At this step, you can right away configure [automatic addition of your cash registers](#) to the personal account of the OFD.

## **initpro requires an EDS**

To register a cash desk through the FTS website, you will need an EDS. If you don't have it, you can [order it from us](#) or submit a paper application for registering the cash register with the tax service.

## **initpro free code**

You can find the free OFD activation code in [cash register profile](#) in the personal account of Initpro.

## **no integration setup required**

You don't need integration setup if:

- you are using [email invoicing service](#);
- you are using [payment link generator](#);
- you are working with the following CMSs: [1C-Bitrix](#), [WordPress](#), [Joomla!](#);
- use the default product transfer from [fiscalization settings](#) of Merchant Portal

Sberbank.

## **closeOfdReceipt**

- A closing receipt can be sent multiple times for one order. There is no limit on the amount, that is, the amount indicated in the closing receipt may be less or greater than the order amount.
- An exception is made for orders for which a refund has been made. For orders for which a refund was made, the closing of the check can only be done if the total of all refunds is less than the confirmed order amount (i.e., there was no full refund).

## **closing if refund**

Closing a check can only be done if a partial, not full refund has been made for the order.

## **amount info**

If you specify amount=0 in the request, a refund is made for the entire amount of the order.

#### **amount credit**

Before summing up all line items, the product of quantity and price for each good item is rounded to an integer. If the value is followed by 5 or greater after the decimal point, it is rounded up. Below there are examples of rounding.

1. If quantity = 0.111, and price = 5500, then the result is 611 (610.5 rounded up).
2. If quantity = 1.455, and price = 6900, then the result is 10040 (10039.5 rounded up).
3. If quantity = 1.211, and price = 6988, then the result is 8462 (8462.468 rounded down).

Therefore, the «amount» parameter value must be equal to the sum of the line items rounded.

#### **credit only rest**

Online Lending functionality is available through the REST interface only.

#### **items credit warning**

Do not use «'» character combination inside this section, or it will cause an error on the gateway side.

#### **itemprice info**

- itemPrice cannot be less than 0.
- itemPrice may be either string or integer.

#### **getreceiptstatus warning**

To execute the request, the order number or its identifier in the fiscalizator must be specified.

#### **customerdetails info credit**

A parameter included in a block and indicated as mandatory is only mandatory within that block. If the block itself is not mandatory and is not passed, its included parameters should not be passed.

#### **the ability to make two-phase payments**

Two-phase payments functionality is available if specifically requested by Merchant from the bank.

#### **preauth parameter**

This parameter can be passed either as a boolean value (without quotation marks) or as a string (with

quotation marks).

#### **email ОФД**

Most OFDs support delivery of receipt only to one email address. Some OFDs allow receipt delivery to multiple emails. In this case, the emails should be separated by commas without spaces. Contact your OFD for availability of receipt delivery to multiple emails.

#### **positionid warning**

The registered cart must contain an item with this positionId value, otherwise the order will fail.

#### **callback delimiter**

Use semicolon («;») without a space as the separator. The string should end with semicolon («;») as well.

#### **sending a notification is considered unsuccessful**

If a response other than the 200 OK HTTP code is returned to the payment gateway, the notification sending is considered to be unsuccessful. In this case, the payment gateway repeats sending the notification (the first attempt is made after 30 seconds, and the next with an interval of 10 minutes) until one of the following conditions is met:

- the payment gateway receives the 200 OK HTTP code in response to the callback notification

or

- there are six unsuccessful attempts to inform in sequence.

When either of the above conditions is met, attempts to send a callback notification about the operation are terminated.

#### **REST authorization**

The transfer of authorization data (username/password) can be transferred both in the request body and in the header.

#### **standard refund and reversal requests**

Refund and reversal requests are standard payment gateway requests.

### **token from tech support**

The token value can be obtained from the technical support staff.

### **autocompletiondate**

To enable this functionality, please contact technical support.

### **itemdetails warning**

The field size limit is 1024 bytes.

### **name (items)**

Use

- to pass \ use \«- to pass »

### **phone note**

If the number is passed in a separate parameter and in additional parameters, the number specified in this phone parameter will be used.

### **customization of the payment page**

Currently, minimal customization of the payment page is available, only the following elements can be changed:

- logo;
- footer.

The requirements for these elements are presented in the instructions for working with your personal account.

### **recurrent payment auto payments**

For recurring payments via SberPay, only one-phase payment can be used. At the same time, for ordinary bindings, a two-phase payment is possible.

### **data types**

In this document, the following data type conventions are used when describing request and response

parameters:

- A<n> - a sequence of Latin letters of length <n>;
- A..<n> - a sequence of Latin letters with a length not exceeding <n>;
- N<n> - a sequence of digits of length <n>;
- N..<n> - a sequence of digits with a length not exceeding <n>;
- AN<n> - a sequence of Latin letters and numbers of fixed length <n>;
- AN.. <n> - a sequence of Latin letters and numbers with a length not exceeding <n>;
- ANS<n> - a sequence of Latin letters, numbers and characters of fixed length <n>
- ANS.. <n> - a sequence of Latin letters, numbers and characters with a length not exceeding <n>;
- UTC - date and time, in this case: the date must be passed without specifying the time zone, Moscow time, for the SOAP protocol, the standard encoding xs: dateTime is used.

## **data types 2**

In this document, the following data type conventions are used when describing request and response parameters:

- string - string;
- array - data array;
- object - data object;
- data - date;
- boolean - boolean (logical) data type;
- int. (integer) - integer data type;
- amount - a numeric data type (amount).

## **ATOL phone**

When using ATOL fiscalization, the phone parameter should be sent in the following format:

The phone number must be passed together with the country code without spaces and additional characters except for the quotation marks (the number «371 2 1234567» must be passed as «+37121234567»). If the phone number belongs to Russia (prefix «+7»), then the value can be passed without a prefix (the number «+79251234567» can be sent as «9251234567»). The maximum string length is 64 characters. One of the fields must be filled in the request: email or phone.

## **reserved names in params, jsonParams, additionalParameters (Spasibo)**

It is forbidden to pass reserved names in the parameter (if they are passed, the order may be rejected):

- sbrf\_spasibo:amount\_bonus
- sbrf\_sbermiles:amount\_bonus
- loyaltyId

## possible values of measure

### Possible values of **measure**

Value	Description
0	Used for payment subjects that can be sold individually or in units (as well as in the event that the payment subject is a product subject to mandatory labeling with an identification tool (which can be known from mark_code being passed))
10	Gram
11	Kilogram
12	Ton
20	Centimeter
21	Decimeter
22	Meter
30	Square centimeter
31	Square decimeter
32	Square meter
40	Milliliter
41	Liter
42	Cubic meter
50	Kilowatt hour
51	Gigacalorie
70	Day (24h)
71	Hour
72	Minute
73	Second
80	Kilobyte
81	Megabyte
82	Gigabyte
83	Terabyte
255	Applicable when using other units of measurement

### markQuantity

Parameters of **markQuantity** tag:

*A parameter included in a tag and indicated as mandatory is only mandatory within that tag. If the tag itself is not mandatory and is not passed, its included parameters should not be passed.*

Name	Type	Mandatory	Description
numerator	N..12	yes	The numerator of the fractional part of the payment subject.
denominator	N..12	yes	The denominator of the fractional part of the payment subject.

The table below shows how the merchant and the payment gateway interact when a payment is refunded.



	<p>The merchant makes a refund request to the buyer:</p> <ul style="list-style-type: none"> <li>• full refund: <ul style="list-style-type: none"> <li>◦ <a href="#">REST interface (refund.do)</a>;</li> <li>◦ <b>1</b> <a href="#">WSDL interface (refundOrder)</a>;</li> </ul> </li> <li>• partial refund: <ul style="list-style-type: none"> <li>◦ <a href="#">REST interface (refund.do)</a>;</li> <li>◦ <a href="#">WSDL interface (refundOrder)</a>.</li> </ul> </li> </ul>
<b>2</b>	<p>After receiving a successful response from the Payment Gateway, the Merchant makes a request for order status:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do)</a>;</li> <li>• <a href="#">WSDL interface (getOrderStatusExtended)</a>.</li> </ul>

There is an option to set up [callback notifications](#) about order status instead of sending the `getOrderStatusExtended` request. These notifications will be sent to you automatically after the order status changes.

## Apple Pay

### Payment using Apple Pay (shopping cart)

On a payment with Apple Pay, the interaction occurs according to the following scheme.

	One-phase payment	Two-phase payment
<b>1</b>	The user selects in mobile app Apple Pay as payment method.	
<b>2</b>	The payment data is sent to the Apple Pay system for processing.	
<b>3</b>	To process payment data in the Apple Pay system, a <code>PKPaymentToken</code> Object is created, which contains the <code>paymentData</code> property (hereinafter, see <a href="#">Apple Pay documentation</a> ).	
<b>4</b>	Apple Pay sends a response to the merchant (mobile app).	
<b>5</b>	The merchant extracts from the received <code>PKPaymentToken</code> Object object the <code>paymentData</code> property and encodes its content in Base64.	

	One-phase payment	Two-phase payment
6	<p>The merchant creates a payment request containing <code>paymentData</code> attribute, which is received from the Apple Pay response and encoded in Base64, and sends it to the payment gateway for processing. The following request is used to process a payment through Apple Pay:</p> <ul style="list-style-type: none"> <li>• <a href="#">payment.do (REST interface)</a>;</li> <li>• <a href="#">applePay (WSDL interface)</a>.</li> </ul> <p>Notably, the <code>preAuth</code> parameter either is not passed or has the value <code>false</code>.</p>	<p>The merchant creates a payment request containing <code>paymentData</code> attribute, which is received from the Apple Pay response and encoded in Base64, and sends it to the payment gateway for processing. The following request is used to process a payment through Apple Pay:</p> <ul style="list-style-type: none"> <li>• <a href="#">payment.do (REST interface)</a>;</li> <li>• <a href="#">applePay (WSDL interface)</a>.</li> </ul> <p>Notably, the <code>preAuth</code> parameter is passed in the request with value <code>true</code>.</p>
7	<p>The payment gateway processes the request and returns a response with the result:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface</a>.</li> <li>• <a href="#">WSDL interface</a>.</li> </ul>	
8	The App displays to the user the result of the payment.	
9	<p>The merchant requests from payment gateway the order payment status by passing a unique identifier that was received upon order registration in <code>orderId</code> parameter:</p> <ul style="list-style-type: none"> <li>• <a href="#">getOrderStatusExtended.do (REST interface)</a>;</li> <li>• <a href="#">getOrderStatusExtended (WSDL interface)</a>.</li> </ul>	
10	Not applicable	<p>To debit funds from the customer's account, the merchant needs to send to the payment gateway the order completion request.</p> <ul style="list-style-type: none"> <li>• <a href="#">deposit.do (REST interface)</a>.</li> <li>• <a href="#">depositOrder (WSDL interface)</a>.</li> </ul>
11	Not applicable	<p>The payment gateway returns the result of processing the request. The order status is not returned. To get the status of the order, an order status request should be sent to the gateway:</p> <ul style="list-style-type: none"> <li>• <a href="#">getOrderStatusExtended.do (REST interface)</a>.</li> <li>• <a href="#">getOrderStatusExtended (WSDL interface)</a>.</li> </ul>

## Google Pay

### Payment using Google Pay (shopping cart)

- Information for developers is available in [Google documentation](#).
- The application must be designed in accordance with [brand guidelines](#).

When submitting a request to Google Pay, use the following parameters:

- gateway: sberbank;
- gatewayMerchantId: <name of your organization in the payment gateway system> .

In case of payment with Google Pay from mobile app, the exchange goes as follows.

	One-phase payment	Two-phase payment
<b>1</b>	The user selects Google Pay as payment method.	
<b>2</b>	The App requests from Google Pay the masked card data.	
<b>3</b>	Google Pay returns masked payment card data to the App.	
<b>4</b>	The application displays the masked card data added to Google Pay to the customer.	
<b>5</b>	The user confirms payment.	
<b>6</b>	The App requests from Google Pay the encrypted card data.	
<b>7</b>	Google encrypts the data using a public key – the corresponding private key is located in the payment gateway.	
<b>8</b>	Google Pay returns encrypted payment data to the App.	
<b>9</b>	<p>The application sends a payment request to the payment gateway, specifying the token received from Google Pay:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (payment.do);</a></li> <li>• <a href="#">WSDL interface (googlePay).</a></li> </ul> <p>Notably, the preAuth parameter either is not passed or has the value false.</p>	<p>The application sends a payment request to the payment gateway, specifying the token received from Google Pay:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (payment.do);</a></li> <li>• <a href="#">WSDL interface (googlePay).</a></li> </ul> <p>Notably, the preAuth parameter is passed in the request with value true.</p>
<b>10</b>	The Payment Gateway decrypts the received token and carries out the payment.	
<b>11</b>	The Payment Gateway returns the payment result to the App.	
<b>12</b>	The App displays the purchase result to the Payer.	
<b>13</b>	<p>The merchant requests from payment gateway the order payment status by passing a unique identifier that was received upon order registration in orderId parameter:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do);</a></li> <li>• <a href="#">WSDL interface (getOrderStatusExtended).</a></li> </ul>	
<b>14</b>	<b>Not applicable</b>	<p>To debit funds from the customer's account, the merchant needs to send to the payment gateway the order completion request:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (deposit.do);</a></li> <li>• <a href="#">WSDL interface (depositOrder).</a></li> </ul>

	One-phase payment	Two-phase payment
15	Not applicable	<p>The payment gateway returns the result of processing the request. The order status is not returned. To get the status of the order, an order status request should be sent to the gateway:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do);</a></li> <li>• <a href="#">WSDL interface (getOrderStatusExtended).</a></li> </ul>

## Samsung Pay

### Payment via Samsung Pay (shopping cart)

In case of payment with Samsung Pay, the exchange goes as follows.

	One-phase payment	Two-phase payment
1	The user selects Samsung Pay as payment method.	
2	The application sends the payment details to Samsung.	
3	Samsung sends to the application a response containing, among other things, the 3ds . data parameter with the encrypted payment data.	
4	<p>The Merchant sends to Payment Gateway a payment request:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (payment.do);</a></li> <li>• <a href="#">WSDL interface (samsungPay).</a></li> </ul> <p>Notably:</p> <ul style="list-style-type: none"> <li>• the paymentToken parameter includes content of 3ds . data received from Samsung;</li> <li>• The preAuth parameter either is not passed or has the value false.</li> </ul>	<p>The Merchant sends to Payment Gateway a payment request:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (payment.do);</a></li> <li>• <a href="#">WSDL interface (samsungPay).</a></li> </ul> <p>Notably:</p> <ul style="list-style-type: none"> <li>• the paymentToken parameter includes content of 3ds . data received from Samsung;</li> <li>• The preAuth parameter is passed in the request with value true.</li> </ul>
5	The Payment Gateway decrypts the received token and carries out the payment.	
6	The Payment Gateway returns to the App the result of payment; the result gets shown on the screen of mobile device.	
7	<p>The merchant requests from payment gateway the order payment status by passing a unique identifier that was received upon order registration in orderId parameter:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do);</a></li> <li>• <a href="#">WSDL interface (getOrderStatusExtended).</a></li> </ul>	
8	Not applicable	<p>To debit funds from the customer's account, the merchant needs to send to the payment gateway the order completion request.</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (deposit.do);</a></li> <li>• <a href="#">WSDL interface (depositOrder).</a></li> </ul>

One-phase payment	Two-phase payment
9 Not applicable	<p>The payment gateway returns the result of processing the request. The order status is not returned. To get the status of the order, an order status request should be sent to the gateway:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do);</a></li> <li>• <a href="#">WS interface (getOrderStatusExtended).</a></li> </ul>

## PPayment with the provision of the possibility of Online Lending

### Integration scheme for using Internet crediting

1	The customer created an order on the merchant's website and selected the «Buy on credit» option.
2	The merchant's system registers the order in the payment gateway: * <a href="#">order registration request</a> or * <a href="#">order registration with fiscalization request</a>
3	The payment gateway passes to the merchant the result of the order registration, including the unique order ID and the URL of the loan term selection page, to which the customer's browser should be redirected to.
4	The merchant redirects the customer's browser to the URL received at the previous step.
5	The customer's browser goes to the specified URL.
6	The customer is presented with a page describing the loan.
7	The customer clicks on <b>Submit application</b> .
8	The payment gateway registers the application.
9	The payment gateway redirects the customer's browser to the Sberbank Online URL.
10	The customer's browser goes to the Sberbank Online authentication page.
11	In the personal account of Sberbank Online, the customer selects the loan term and undergoes the loan issuance procedure.
12	Sberbank Online shares data with the payment gateway.
13	Sberbank Online makes payment.
14	Sberbank Online informs RBS about changing the order status to «Paid».
15	After making the payment, Sberbank Online redirects the customer to the merchant 's page.
16	Payment gateway communicates with OFD (Fiscal Data Operator).
17	The payment gateway updates order status.
18	The merchant sends an <a href="#">order status request</a> to the payment gateway.
19	The payment gateway returns a response to the order status request.

There is an option to set up [callback notifications](#) about order status instead of sending the getOrderStatusExtended request. These notifications will be sent to you automatically after the order status changes.

### Payment reversal

Reversal of a payment for an order is available to Merchants provided that they have the

corresponding permissions and approval of Sberbank.

- With one-phase payments, payment reversal is possible for orders in the state **Deposited**, but only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.
- In case of two-phase payments, a payment reversal can be processed for orders in the **Approved** status.

The table below shows how the merchant and the payment gateway interact when a payment is reversed.

1	The merchant makes a request to reverse the payment: <ul style="list-style-type: none"><li>• <a href="#">REST interface (reverse.do)</a>;</li><li>• <a href="#">WSDL interface (reverseOrder)</a>.</li></ul>
2	After receiving a successful response from the Payment Gateway, the Merchant makes a request for order status: <ul style="list-style-type: none"><li>• <a href="#">REST interface (getOrderStatusExtended.do)</a>;</li><li>• <a href="#">WSDL interface (getOrderStatusExtended)</a>.</li></ul>

There is an option to set up [callback notifications](#) about order status instead of sending the `getOrderStatusExtended` request. These notifications will be sent to you automatically after the order status changes.

## Refunds to the payer

The Payment Gateway allows full or partial refunds for paid orders.

### callback instead of getOrderStatusExtended

There is an option to set up [callback notifications](#) about order status instead of sending the `getOrderStatusExtended` request. These notifications will be sent to you automatically after the order status changes.

### getreceiptstatus

To get information about a sales receipt (for example, to then pass this data to a customer) use the [getReceiptStatus.do](#) request. Currently, this request can only be made through the REST interface.

### recommended wordpress version

Recommended version of WordPress for working with the Sberbank payment plugin: not lower than

4.8.3.

### **download woocommerce**

You can download the required version of WooCommerce on [github.com](#) (link leads to WooCommerce version history).

### **woocommerce compatibility**

The WooCommerce plugin is not compatible with all WordPress themes. Here, the [Storefront](#) theme is used as example. This theme is designed by WooCommerce developers.

### **meanings of terms**

Meanings of terms highlighted on this page in **bold** font are given in the [glossary](#).

### **types of payments**

One-phase payments are preferable if the product or service is provided immediately after payment. Two-phase payments should be used if some time elapses between the buyer's decision to pay and the delivery of the selected product or service.

### **need to check 3dsec**

The need for verification using 3-D Secure depends on the buyer's bank card, as well as on the settings configured for the Merchant upon agreement with Sberbank.

### **reversal before midnight**

In case of one-phase payments, the reversal operation is possible only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.

### **access to personal account**

Regardless of the chosen connection method, you are given access to the [personal account](#) on the payment gateway server.

### **you can use test cards**

You can use [test cards](#) to try out the functionality.

## **access to payment methods**

The available payment acceptance methods depend on settings configured for you. If in doubt, check with a Sberbank representative which method of accepting payments you can use.

## **passed as clientid**

The value of this parameter goes in the registration request as the `clientId` parameter.

## **need to use utf-8**

The page of your site on which the payment solution will be used must be saved in UTF-8 encoding, and it must also have a meta-tag `<meta name=«viewport» content=«width=device-width»>`.

## **update the links**

The links point to test environment. Once done with testing, update the links so that they point to the production environment.

## **tls is required**

The server on which the content management system is located must support the TLS 1.2 protocol. If in doubt, check with your hosting provider.

## **minimum amount to be debited**

The minimum amount of funds to be debited is 100 rubles.

## **default parameters**

If the cart and fiscalization data are not passed in the request, the default values specified in the personal account are passed to the Fiscal Data Operator (see [setting up fiscalization data](#) for more details).

## **passing a field to processing**

To enable sending this field to the processing system, contact the technical support service.



## **parameter must be specified**

Specifying this parameter when processing payments with the use of bindings is mandatory. Otherwise, a payment will be unsuccessful.

## **fields are passed by default**

By default, the following fields are passed to the Bank processing system:

- `orderNumber` – order number in the store system;
- `description` – order description (no more than 24 characters; %, +, carriage return \r, and line feed \n cannot be used).

## **mandatory to pass either email or phone**

It is mandatory to pass one of the two parameters: email or phone.

## **parameters are not required if the block is missing**

A parameter included in a block and indicated as mandatory is only mandatory within that block. If the section is optional and missing, then the parameters included in it should not be passed.

## **access to personal account**

Access to personal account is granted regardless of the chosen connection method.

## **access to functions depends on settings**

Functions available in the personal account depend on the configured settings. If in doubt, please contact a Sberbank representative.

## **more details available on the 1c-bitrix website**

For more details on the differences between editions, see the website «[1C-Bitrix](#)».

## **1c-bitrix version used as example**

1C-Bitrix version used for installation and configuration: **1C-Bitrix: Site Manager, Small Business Edition**, version **16.5.4**.

## terms in the 1c-bitrix interface

In the 1C-Bitrix interface, a «payment system» is a payment method that a customer can choose, for example: «in cash to a courier», «by bank card», etc. The term «payment method» is used throughout this description.

## 1C-Bitrix settings order

The order of fields and settings may differ depending on the edition of your 1C-Bitrix instance.

## 1C-Bitrix using default tab as example

In this description, the setting is shown using the example of the **Default** tab. If the tabs **Individual** and **Entity** have not been edited before, the changes made will apply to these tabs.

## password if the plugin is in test mode

If the plugin is running in test mode, you should specify the password for the test user account. If the plugin is running in standard (production) mode, you should specify the password for the production user account.

## login if the plugin is in test mode

If the plugin is running in test mode, you should specify the login for the test user account. If the plugin is running in production mode, you should specify the password for the production user account.

## wordpress display order

If you want to change the order in which a payment method is displayed for a customer, click on the icon opposite the required payment method and, while holding the left mouse button, move this payment method to the required position.

## cs-cart compatibility

The payment plugin was tested for compatibility with CS-Cart version 4.6.x.

## terms in the cs-cart interface

In the CS-Cart interface, modules that extend the site's functionality are called «add-ons». In this description, the term «plugin» is used.

## **plugin settings only**

The description contains only those settings that relate directly to the payment plugin. The rest of the settings are related to the content management system (CMS).

## **terms in the Drupal interface**

In the Drupal interface, add-ons are called «modules». The payment plugin falls into the same category, however the term «plugin» is used in this description.

## **drupal description based on example**

A combined distribution [Commerce Kickstart](#) version 7.x-2.45 was used for the purposes of this description. This distribution combines Drupal version 7.54 and Drupal Commerce version 7.x-1.13.

## **opencart compatibility**

The plugin was tested on OpenCart/ocStore version 2.3.0.2.

## **the location of the log files**

The event log files are located here: <site folder>/system/storage/logs/

## **prestashop compatibility**

The instructions for installing and configuring the plugin for PrestaShop version 1.7.x are given as example.

## **netcat compatibility**

The plugin is compatible with Netcat versions 5.9.x and higher and has been tested on E-commerce editions.

## **avoid error in personal account when testing REST**

This situation can be avoided in one of the following ways:

- use the Incognito mode for REST requests, and work in your personal account in the browser window as usual;
- use different browsers to test REST requests and work in your personal account.

## wsdl connection URLs

Use the following URL to connect to the Test Service (WSDL):

<https://3dsec.sberbank.ru/payment/webservices/merchant-ws?wsdl>.

## payment reversal conditions

- With one-phase payments, payment reversal is possible for orders in the state **Deposited**, but only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.
- In case of two-phase payments, a payment reversal can be processed for orders in the **Approved** status.

## refund conditions

Refunds are possible for payments in **Deposited** status.

## passing clientId in payment by binding

When `clientId` parameter is passed initially, the client's identifier is registered in the payment gateway and the data of their card is saved (except for the CVC code). On subsequent passes of this parameter, the client is redirected to a page where they can choose from the following options:

- use the previously saved data of the payment card – in this case, the client does not need to re-enter the data of the payment card for payment;
- enter the data of the new payment card – in this case, the data of the new payment card will be additionally linked to the identifier of this client.

## what is a callback

Callback notifications are not email notifications or telephone notifications. These are notifications received through the programming interface.

## contact support to enable callback

To enable the ability to receive checksum notifications and get a private key, contact technical support.

## self-signed certificates cannot be used

Self-signed certificates are not allowed. The certificate must be signed by a trusted certification authority (see above).

## **basic callback parameters**

The table shows the basic parameters. A number of additional parameters that will be passed in notifications can be specified in personal account.

## **parameters must be sorted alphabetically**

Notably, the «parameter\_name;parameter\_value» pairs must be sorted in direct alphabetical order by parameter name.

## **checksum check value**

If the checksums match, this notification is authentic and was sent by the payment gateway. Otherwise, it is likely that the attacker is trying to pass off their notification as a payment gateway notification.

## **use Latin characters**

Use Latin characters when filling out the form.

## **need an apple developer account**

To be able to complete this procedure, you need to have an Apple Developer account.

## **apple pay id creation rules**

The identifier should begin with the word **merchant**. The address of your live site should be specified in the reverse order. For example, for the site **sale.test.ru** the identifier will look as **merchant.ru.test.sale**.

## **checklist for testing apple pay**

Below there is a short list of actions that will allow you to test Apple Pay. Further details are available [in documentation](#) on the Apple website.

## **how to create a test account with apple**

For more information about creating a test account, see [Apple documentation](#).

## **create a new apple account**

If you mistakenly use a test account on a test device to log in to a production environment (such as iTunes) instead of logging in to a test environment, this test account will become invalid and cannot be used again. In this case create a new test account with a new email address.

## **samsung only for payment via mobile app**

Steps 1-3 are necessary to enable payment via the mobile app.

## **it is not necessary to wait for the completion of the report**

You can start generating a new report without waiting for the completion of the previous one.

## **number of generated reports**

Number in brackets after the caption **Reports** means how many reports are currently being generated.

## **fiscalization settings are set by support**

Fiscalization settings are set by technical support upon request to support@ecom.sberbank.ru. At this stage, you do not need to configure the fiscalization parameters yourself.

## **information about actions in atol personal account**

Detailed information about actions in the ATOL personal account is given on [ATOL website](#) (external link).

## **test data for connection to OFD**

Do not specify actual data to connect to the fiscal data operator in the test personal account! If you do so, test payments will be registered with the fiscal data operator as if they were real. After entering the data into the live personal account – check that they are valid and saved correctly.

## **passing empty compositecompletionamount**

If zero is passed in this parameter, completion will occur for the entire pre-authorized amount.

## **required parameters in getOrderstatusextended**

Either orderId or orderNumber must be present in the request. If both parameters are present in the request, orderId has priority.

## **required parameters in getOrderstatusextended ws**

The request must contain either orderId or merchantordernumber. If both parameters are present in the request, orderId has priority.

## **post only**

Only POST is supported.

## **the response depends on the OFD**

Some parameters to be returned may depend on the Fiscal Data Operator used.

## **use standard requests**

Use the standard requests to the payment gateway for the operations of reversal, refund and payment completion.

## **you need to pass the sbrf\_spasibo parameter**

If the merchant is connected to the «Spasibo from Sberbank» affiliate program, and bonus points are sent in the registration request, then the parameter should be passed as an additional parameter sbrf\_spasibo:amount\_bonus indicating the amount of bonus points to be deducted, as in the example below.

```
jsonParams={«sbrf_spasibo:amount_bonus»:«100»}
```

In this case, the amount in money will be deducted from the amount parameter.

## **you need to pass the sbrf\_spasibo parameter (ws)**

If the merchant is connected to the «Spasibo from Sberbank» affiliate program, and bonus points are sent in the registration request, then the parameter should be passed as an additional parameter sbrf\_spasibo:amount\_bonus indicating the amount of bonus points to be deducted, as in the example below.

```
<params name=«sbrf_spasibo:amount_bonus» value=«1030»/>
```

In this case, the amount in money will be deducted from the amount parameter.

### **reversals and refunds cannot be made**

Orders that initiate recurring payments cannot be reversed and refunded, as in these cases there is no real debiting.

### **passing an empty amount**

If you do not specify the amount parameter, completion will take place for the entire pre-authorized amount.

### **quantity and positionid are sufficient**

For error-free processing of the request it is sufficient to pass the quantity and positionId parameters.

### **passing empty depositamount**

If you do not specify the depositAmount parameter, completion will take place for the entire pre-authorized amount.

### **authcode field is obsolete**

The order status must be determined by the orderStatus parameter value. The authCode field is obsolete.

### **getreceiptstatus is not supported in ws**

Currently, requesting information about the status of a check in the WS interface is not supported.

### **download virtuemart**

From the VirtueMart developer site you can [download](#) the Joomla! distribution with pre-installed add-on VirtueMart. The names of such combined archives end with **Stable-Full\_Package.zip**. When doing this, choose a combination of Joomla! and VirtueMart, compatible with the payment plugin.

### **joomla compatibility**

The plugins are only guaranteed to work with the following combinations of Joomla! and VirtueMart



add-ons. If you use other combinations of versions (for example, after updating Joomla!) The plugin may not work correctly.

### **compliance with 54-fz**

Starting from this version, the plugin allows you to transfer a shopping cart, which provides the possibility of detailed fiscalization in accordance with [54-FZ](#).

### **on the example of the next version of joomla**

Installing and configuring the plugin is presented on Joomla! version 3.6.5 and VirtueMart version 3.2.0.

### **android pay not supported**

The Android Pay payment method is currently not supported by Google. This payment method will be replaced by Pay with Google (it is currently temporarily unavailable for integration, documentation is under development).

### **section in development**

The section is currently under construction.

### **itemamount is rounded to the second decimal place**

When calculating  $\text{itemAmount} = \text{itemPrice} * \text{quantity}$  parameter, the result is rounded up to the second digit after the decimal point. For example, if the calculation result is 100.255, then final value will be 100.26.

### **orderid ordernumber or merchantlogin required**

To execute the request, one of the following parameters must be specified:

- 2;
- 2;
- 2.

### **access to payment method**

To be able to accept payment in this way, you must have the respective rights in the system. If in doubt, contact technical support.

## **only one-phase payments**

Payment in this way is possible only when using a one-phase payment scheme.

## **asymmetric cryptography is recommended**

For greater security, use of a method in which the checksum is generated using asymmetric cryptography is recommended.

## **the order of the parameters can be arbitrary**

The order of parameters in the notification can be undefined.

## **symmetric cryptography example**

Below are examples of creating a checksum using symmetric cryptography.

## **in order to familiarize**

The information on this portal is provided as an introduction to the functionality of the payment gateway. The necessary data for connecting to the live environment will be provided after testing in accordance with [connection checklist](#).

## **request only through rest**

Currently this request is available through REST interface only.

## **for samsung pay web you need a javascript script**

To be able to receive payment via Samsung Pay Web with a merchant-side payment page, you must install [JavaScript script](#). When receiving a response from the payment gateway, you need to call the connect function of this script and pass in it certain parameters of the payment gateway response. For details, see the description of the corresponding [interaction schemes](#) and description of the [request](#).

## **https and tls are required**

Sites must use the HTTPS scheme and support the TLS 1.2 protocol.

## **safari required**

When making a payment from a web page, the buyer must be in the Safari browser.

## **programming skills required**

Programming skills are required to implement automatic check of payment completion.

## **don't share the private key with anyone**

Do not share or publish the private key to anyone. Signature verification must be implemented on the store's server. JavaScript implementation of signature verification on the front-end of the store is not allowed.

## **you need to have google merchant id**

To make payments from the website, your online store site must be registered with Google Pay.

## **for ofd, apply for connection**

Before registering a personal account of a cloud cash register provider [apply](#) to connect to Internet acquiring.

## **additionalparameters**

If the merchant has fiscalization set up, when indicating email (the buyer's email address) and/or phone (the buyer's cell phone number) as additional parameters, these parameters are first used to send the fiscal receipt.

## **test and live mode for plugins**

- If the plugin is running in test mode, in the respective fields (**Login** and **Password**) you should specify the data of the test service account.
- If the plugin works in the live mode, in the respective fields (**Login** and **Password**) you should specify the data of the live service account.

## **only if fiscalization is enabled**

This setting is applied only if you have configured fiscalization parameters – see details in [section on compliance with law 54-FZ](#).

## **fiscalization support for amiro**

Fiscalization is supported in plugins for Amiro.CMS versions 7.x and later (see [details on fiscalization](#)).

### **only for amiro 7 and higher**

The setting is available only in plugins for Amiro.CMS versions 7.x and later.

### **hold ctrl to select multiple items**

If you want to select multiple items, hold down the CTRL key on your keyboard before checking the items in the list.

### **plugin supports fiscalization**

The plugin supports fiscalization in accordance with the law [54-FZ](#).

### **features verify**

- Even if the payment amount is to be passed in the request, it will not be debited from the account.
- After the order has been successfully registered, it is put to the REVERSED (cancelled) status.

### **integration is not needed under the following conditions**

Integration with a payment gateway is not required if you use:

- [e-mail invoicing service](#);
- [payment links generator](#);
- work using [Yoom.Shop](#) (external ink);
- work using the following CMS:
  - [1C-Bitrix](#);
  - [WordPress](#);
  - [Joomla!](#);
- use the transfer of goods by default from the fiscalization settings of the Sberbank personal account.

### **checking fiscalization on the live server**

After making all the settings, we ask you to carefully check the correct filling of all fields in relation to the fiscalization parameters in the personal account of the bank, because the correctness of the transfer of fiscal data to the OFD and tax depends on this. We also recommend checking «in action» when sending a fiscal receipt.

In case of any difficulties or questions, you can send them to the address of the technical support of the bank [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru) and Business.Ru [support@mail.business.ru](mailto:support@mail.business.ru).

## **google pay developer info**

Information for developers is available in [Google documentation](#).

## **information for Google Pay Web developers**

- Information for developers is available in [Google documentation](#).
- The payment page must be designed in accordance with [brand guidelines](#).

## **information for Google Pay In-App developers**

- Information for developers is available in [Google documentation](#).
- The application must be designed in accordance with [brand guidelines](#).

## **google tos**

By accepting payments via Google Pay, you agree to [Google terms of service](#).

## **passing the shopping cart is optional**

If the cart and fiscalization data are not included in the request, then the default values specified in the personal account are passed to the Fiscal Data Operator (see User Manual for personal account).

## **rest request type**

Using the REST protocol, you should send requests with the type `application/x-www-form-urlencoded`, rather than `multipart/form-data`.

## **not enough rights to submit a payment form**

If the section **Invoice to email** the following message is displayed: "You do not have sufficient rights to send the payment form", please contact the technical support service [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru) with a request to activate the invoicing option. Make sure to include the following information in your request:

- operator login (used to enter your personal account).
- environment (test/production) for which changes need to be made

## **QR code for payment link**

The link can also be encoded as a QR code.

## **closing Merchant Portal session**

If there are no user actions for **10+ minutes**, the session is automatically closed.

## **exporting charts from Merchant Portal**



A click on  saves the chart image.

## **you can select several statuses**

You can set several statuses at the same time by pressing CTRL and selecting them with a left-click.

## **short payment term**

If you set the invoice validity period too short, the client will not be able to use the same invoice to pay for the order after some time.

## **use of two-phase payment**

When using a two-phase payment, it is recommended to debit the amount no later than 7 calendar days (for some industries, such as the hotel business, car rental, etc., the period of possible holding may be increased).

## **change password for production**

Passwords provided for both logins in production environment must be changed in the personal account (available [here](#)).

## **make sure to change password for all users**

Password for each login must be changed.

## **two-phase payment**

When using **two-phase** payment, it is recommended to debit the amount no later than 7 calendar days (for some industries, such as the hotel business, car rental, etc., the period of possible holding may be increased).

## **plugins support 54-fz**

### **Fiscalization - 54-FZ**

Most of the plugins presented support fiscalization. To clarify whether the plugin you are interested in supports the fiscalization functionality, you can visit our technical support address – [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru).

## **support for plugins**

We are ready to develop plugin support for new CMS. To do this, you need to contact the technical support service – [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru).

## **payment phases in Online Lending**

For online lending, only one-phase payment is applied.

### **only one-phase payment**

When using the Online Lending functionality, only the one-phase payment is available.

### **token authentication not possible**

When using the Online Lending functionality, token authentication is not possible. Login and password should be used for authentication.

### **only through rest**

Online Lending functionality is available through the REST interface only.

### **itemamount gift**

If all the items in the request are delivered without payment (a gift to customer), the value 0 should be passed.

## **itemprice gift**

If the item is delivered without payment (a gift to customer), the value 0 should be passed.

## **can be changed in refund**

When requesting a refund, the value of this parameter may differ from that given during the registration of the order.

## **passing additionalOfdParams**

The transfer of this block is possible only when using the following OFDs:

- ATOL;
- Business.Ru;
- Evotor.

## **ffd 105 and higher**

The settings are effective only if the format of fiscal documents of version 1.05 and higher is selected.

## **FFD version must match**

The format of the version must match the format selected in the personal account of the bank and in the account of the fiscalization service.

## **add json header**

It is necessary to add to the request a header with the definition of the type of contents – «Content-Type: application/json» to ensure that the request is processed correctly.

## **assumes joomshopping is installed**

This tutorial assumes that your Joomla! the JoomShopping component is already installed.

## **assumes virtuemart is installed**

This tutorial assumes that your Joomla! the VirtueMart component is already installed.



## **simpla version**

The plugin was tested for Simpla version 3.2.x.

## **google pay registration**

These steps are necessary if you want to receive encrypted payment actions directly from Google Pay and redirect them to a payment gateway. Namely:

- online payment, if the payment page is located on the merchant's side;
- payment from the application.

## **google pay info**

For more information see [Integration Checklist](#) (external link).

## **google pay create account**

It is recommended to use an existing email address on your domain, for this you need to select the option **Use current email address**.

## **samsung pay registration**

These actions are necessary to enable payment via the mobile app. You do not need to take these steps to pay through the site.

## **shop-script version**

Shop-Script version must be 7.x or higher.

## **not for ofd.ru**

Not applicable for **OFD.ru**.

## **magento compatibility**

Magento plugins for versions 2.2.x and 2.3.x are incompatible. When upgrading Magento version from 2.2.x to 2.3.x, you must first uninstall the plugin for version 2.2.x and then install the plugin for version 2.3.x.

## **php folder**

The beginning of the command depends on the folder where PHP is installed. In particular, the PHP version can be specified without a hyphen: php5.6, php7.1, php7.2, etc.

If in doubt, contact your hosting provider.

## **may not have enough memory**

On shared hosting, the last command may not be executed due to the memory limit set by the hosting provider. In this case, contact your hosting provider.

## **uninstalling magento**

In code folder, as well as in the folders of a higher level, (app) there may be data not related to the plugin, so delete only the Rbs folder and its contents.

## **6 and 8 are not used**

Types 6 (gambling winnings) and 8 (lottery winnings) are not used.

## **initpro start of integration**

At this step, you can right away configure [automatic addition of your cash registers](#) to the personal account of the OFD.

## **initpro requires an EDS**

To register a cash desk through the FTS website, you will need an EDS. If you don't have it, you can [order it from us](#) or submit a paper application for registering the cash register with the tax service.

## **initpro free code**

You can find the free OFD activation code in [cash register profile](#) in the personal account of Initpro.

## **no integration setup required**

You don't need integration setup if:

- you are using [email invoicing service](#);
- you are using [payment link generator](#);

- you are working with the following CMSs: [1C-Bitrix](#), [WordPress](#), [Joomla!](#);
- use the default product transfer from [fiscalization settings](#) of Merchant Portal

Sberbank.

### **closeOfdReceipt**

- A closing receipt can be sent multiple times for one order. There is no limit on the amount, that is, the amount indicated in the closing receipt may be less or greater than the order amount.
- An exception is made for orders for which a refund has been made. For orders for which a refund was made, the closing of the check can only be done if the total of all refunds is less than the confirmed order amount (i.e., there was no full refund).

### **closing if refund**

Closing a check can only be done if a partial, not full refund has been made for the order.

### **amount info**

If you specify amount=0 in the request, a refund is made for the entire amount of the order.

### **amount credit**

Before summing up all line items, the product of quantity and price for each good item is rounded to an integer. If the value is followed by 5 or greater after the decimal point, it is rounded up. Below there are examples of rounding.

1. If quantity = 0.111, and price = 5500, then the result is 611 (610.5 rounded up).
2. If quantity = 1.455, and price = 6900, then the result is 10040 (10039.5 rounded up).
3. If quantity = 1.211, and price = 6988, then the result is 8462 (8462.468 rounded down).

Therefore, the «amount» parameter value must be equal to the sum of the line items rounded.

### **credit only rest**

Online Lending functionality is available through the REST interface only.

### **items credit warning**

Do not use «')» character combination inside this section, or it will cause an error on the gateway side.

## **itemprice info**

- itemPrice cannot be less than 0.
- itemPrice may be either string or integer.

## **getreceiptstatus warning**

To execute the request, the order number or its identifier in the fiscalizator must be specified.

## **customerdetails info credit**

A parameter included in a block and indicated as mandatory is only mandatory within that block. If the block itself is not mandatory and is not passed, its included parameters should not be passed.

## **the ability to make two-phase payments**

Two-phase payments functionality is available if specifically requested by Merchant from the bank.

## **preauth parameter**

This parameter can be passed either as a boolean value (without quotation marks) or as a string (with quotation marks).

## **email ОФД**

Most OFDs support delivery of receipt only to one email address. Some OFDs allow receipt delivery to multiple emails. In this case, the emails should be separated by commas without spaces. Contact your OFD for availability of receipt delivery to multiple emails.

## **positionid warning**

The registered cart must contain an item with this positionId value, otherwise the order will fail.

## **callback delimiter**

Use semicolon («;») without a space as the separator. The string should end with semicolon («;») as well.

## **sending a notification is considered unsuccessful**

If a response other than the 200 OK HTTP code is returned to the payment gateway, the notification sending is considered to be unsuccessful. In this case, the payment gateway repeats sending the notification (the first attempt is made after 30 seconds, and the next with an interval of 10 minutes) until one of the following conditions is met:

- the payment gateway receives the 200 OK HTTP code in response to the callback notification

or

- there are six unsuccessful attempts to inform in sequence.

When either of the above conditions is met, attempts to send a callback notification about the operation are terminated.

## **REST authorization**

The transfer of authorization data (username/password) can be transferred both in the request body and in the header.

## **standard refund and reversal requests**

Refund and reversal requests are standard payment gateway requests.

## **token from tech support**

The token value can be obtained from the technical support staff.

## **autocomplete**

To enable this functionality, please contact technical support.

## **item details warning**

The field size limit is 1024 bytes.

## **name (items)**

Use

- to pass \ use \«- to pass »

## **phone note**

If the number is passed in a separate parameter and in additional parameters, the number specified in this phone parameter will be used.

## **customization of the payment page**

Currently, minimal customization of the payment page is available, only the following elements can be changed:

- logo;
- footer.

The requirements for these elements are presented in the instructions for working with your personal account.

## **recurrent payment auto payments**

For recurring payments via SberPay, only one-phase payment can be used. At the same time, for ordinary bindings, a two-phase payment is possible.

## **data types**

In this document, the following data type conventions are used when describing request and response parameters:

- A<n> – a sequence of Latin letters of length <n>;
- A..<n> – a sequence of Latin letters with a length not exceeding <n>;
- N<n> – a sequence of digits of length <n>;
- N..<n> – a sequence of digits with a length not exceeding <n>;
- AN<n> – a sequence of Latin letters and numbers of fixed length <n>;
- AN..<n> – a sequence of Latin letters and numbers with a length not exceeding <n>;
- ANS<n> – a sequence of Latin letters, numbers and characters of fixed length <n>
- ANS..<n> – a sequence of Latin letters, numbers and characters with a length not exceeding <n>;
- UTC – date and time, in this case: the date must be passed without specifying the time zone, Moscow time, for the SOAP protocol, the standard encoding xs: dateTime is used.

## **data types 2**

In this document, the following data type conventions are used when describing request and response parameters:

- string – string;
- array – data array;
- object – data object;
- data – date;
- boolean – boolean (logical) data type;

- int. (integer) – integer data type;
- amount – a numeric data type (amount).

## ATOL phone

When using ATOL fiscalization, the phone parameter should be sent in the following format:

The phone number must be passed together with the country code without spaces and additional characters except for the quotation marks (the number «371 2 1234567» must be passed as «+37121234567»). If the phone number belongs to Russia (prefix «+7»), then the value can be passed without a prefix (the number «+79251234567» can be sent as «9251234567»). The maximum string length is 64 characters. One of the fields must be filled in the request: email or phone.

## reserved names in params, jsonParams, additionalParameters (Spasibo)

It is forbidden to pass reserved names in the parameter (if they are passed, the order may be rejected):

- sbrf\_spasibo:amount\_bonus
- sbrf\_sbermiles:amount\_bonus
- loyaltyId

## possible values of measure

Possible values of **measure**

Value	Description
0	Used for payment subjects that can be sold individually or in units (as well as in the event that the payment subject is a product subject to mandatory labeling with an identification tool (which can be known from mark_code being passed))
10	Gram
11	Kilogram
12	Ton
20	Centimeter
21	Decimeter
22	Meter
30	Square centimeter
31	Square decimeter
32	Square meter
40	Milliliter
41	Liter
42	Cubic meter
50	Kilowatt hour
51	Gigacalorie
70	Day (24h)
71	Hour

Value	Description
72	Minute
73	Second
80	Kilobyte
81	Megabyte
82	Gigabyte
83	Terabyte
255	Applicable when using other units of measurement

## markQuantity

Parameters of **markQuantity** tag:

*A parameter included in a tag and indicated as mandatory is only mandatory within that tag. If the tag itself is not mandatory and is not passed, its included parameters should not be passed.*

Name	Type	Mandatory	Description
numerator	N..12	yes	The numerator of the fractional part of the payment subject.
denominator	N..12	yes	The denominator of the fractional part of the payment subject.

The table below shows how the merchant and the payment gateway interact when a payment is refunded.

1	<p>The merchant makes a refund request to the buyer:</p> <ul style="list-style-type: none"> <li>• full refund: <ul style="list-style-type: none"> <li>◦ <a href="#">REST interface (refund.do)</a>;</li> <li>◦ <a href="#">WSDL interface (refundOrder)</a>;</li> </ul> </li> <li>• partial refund: <ul style="list-style-type: none"> <li>◦ <a href="#">REST interface (refund.do)</a>;</li> <li>◦ <a href="#">WSDL interface (refundOrder)</a>.</li> </ul> </li> </ul>
2	<p>After receiving a successful response from the Payment Gateway, the Merchant makes a request for order status:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do)</a>;</li> <li>• <a href="#">WSDL interface (getOrderStatusExtended)</a>.</li> </ul>

There is an option to set up [callback notifications](#) about order status instead of sending the getOrderStatusExtended request. These notifications will be sent to you automatically after the order status changes.

## Payment by binding



## Payment by binding (the payment form is located on the server of the payment gateway)

	One-phase payment	Two-phase payment
1	The payer chooses goods or services on the merchant website and picks bank card as a payment option.	
2		

## Payment reversal

Reversal of a payment for an order is available to Merchants provided that they have the corresponding permissions and approval of Sberbank.

- With one-phase payments, payment reversal is possible for orders in the state **Deposited**, but only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.
- In case of two-phase payments, a payment reversal can be processed for orders in the **Approved** status.

The table below shows how the merchant and the payment gateway interact when a payment is reversed.

	The merchant makes a request to reverse the payment:
1	<ul style="list-style-type: none"><li>• <a href="#">REST interface (reverse.do)</a>;</li><li>• <a href="#">WSDL interface (reverseOrder)</a>.</li></ul>
2	<p>After receiving a successful response from the Payment Gateway, the Merchant makes a request for order status:</p> <ul style="list-style-type: none"><li>• <a href="#">REST interface (getOrderStatusExtended.do)</a>;</li><li>• <a href="#">WSDL interface (getOrderStatusExtended)</a>.</li></ul>

There is an option to set up [callback notifications](#) about order status instead of sending the `getOrderStatusExtended` request. These notifications will be sent to you automatically after the order status changes.

## Refunds to the payer

The Payment Gateway allows full or partial refunds for paid orders.

### callback instead of getOrderStatusExtended

There is an option to set up [callback notifications](#) about order status instead of sending the `getOrderStatusExtended` request. These notifications will be sent to you automatically after the order status changes.

## **getreceiptstatus**

To get information about a sales receipt (for example, to then pass this data to a customer) use the [getReceiptStatus.do](#) request. Currently, this request can only be made through the REST interface.

## **recommended wordpress version**

Recommended version of WordPress for working with the Sberbank payment plugin: not lower than 4.8.3.

## **download woocommerce**

You can download the required version of WooCommerce on [github.com](#) (link leads to WooCommerce version history).

## **woocommerce compatibility**

The WooCommerce plugin is not compatible with all WordPress themes. Here, the [Storefront](#) theme is used as example. This theme is designed by WooCommerce developers.

## **meanings of terms**

Meanings of terms highlighted on this page in **bold** font are given in the [glossary](#).

## **types of payments**

One-phase payments are preferable if the product or service is provided immediately after payment. Two-phase payments should be used if some time elapses between the buyer's decision to pay and the delivery of the selected product or service.

## **need to check 3dsec**

The need for verification using 3-D Secure depends on the buyer's bank card, as well as on the settings configured for the Merchant upon agreement with Sberbank.

## **reversal before midnight**

In case of one-phase payments, the reversal operation is possible only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.

## **access to personal account**

Regardless of the chosen connection method, you are given access to the [personal account](#) on the payment gateway server.

## **you can use test cards**

You can use [test cards](#) to try out the functionality.

## **access to payment methods**

The available payment acceptance methods depend on settings configured for you. If in doubt, check with a Sberbank representative which method of accepting payments you can use.

## **passed as clientid**

The value of this parameter goes in the registration request as the `clientId` parameter.

## **need to use utf-8**

The page of your site on which the payment solution will be used must be saved in UTF-8 encoding, and it must also have a meta-tag `<meta name=«viewport» content=«width=device-width»>`.

## **update the links**

The links point to test environment. Once done with testing, update the links so that they point to the production environment.

## **tls is required**

The server on which the content management system is located must support the TLS 1.2 protocol. If in doubt, check with your hosting provider.

## **minimum amount to be debited**

The minimum amount of funds to be debited is 100 rubles.

## **default parameters**

If the cart and fiscalization data are not passed in the request, the default values specified in the personal account are passed to the Fiscal Data Operator (see [setting up fiscalization data](#) for more details).

### **passing a field to processing**

To enable sending this field to the processing system, contact the technical support service.

### **parameter must be specified**

Specifying this parameter when processing payments with the use of bindings is mandatory. Otherwise, a payment will be unsuccessful.

### **fields are passed by default**

By default, the following fields are passed to the Bank processing system:

- `orderNumber` – order number in the store system;
- `description` – order description (no more than 24 characters; %, +, carriage return \r, and line feed \n cannot be used).

### **mandatory to pass either email or phone**

It is mandatory to pass one of the two parameters: `email` or `phone`.

### **parameters are not required if the block is missing**

A parameter included in a block and indicated as mandatory is only mandatory within that block. If the section is optional and missing, then the parameters included in it should not be passed.

### **access to personal account**

Access to personal account is granted regardless of the chosen connection method.

### **access to functions depends on settings**

Functions available in the personal account depend on the configured settings. If in doubt, please contact a Sberbank representative.

## more details available on the 1c-bitrix website

For more details on the differences between editions, see the website «[1C-Bitrix](#)».

## 1c-bitrix version used as example

1C-Bitrix version used for installation and configuration: **1C-Bitrix: Site Manager, Small Business Edition**, version **16.5.4**.

## terms in the 1c-bitrix interface

In the 1C-Bitrix interface, a «payment system» is a payment method that a customer can choose, for example: «in cash to a courier», «by bank card», etc. The term «payment method» is used throughout this description.

## 1C-Bitrix settings order

The order of fields and settings may differ depending on the edition of your 1C-Bitrix instance.

## 1C-Bitrix using default tab as example

In this description, the setting is shown using the example of the **Default** tab. If the tabs **Individual** and **Entity** have not been edited before, the changes made will apply to these tabs.

## password if the plugin is in test mode

If the plugin is running in test mode, you should specify the password for the test user account. If the plugin is running in standard (production) mode, you should specify the password for the production user account.

## login if the plugin is in test mode

If the plugin is running in test mode, you should specify the login for the test user account. If the plugin is running in production mode, you should specify the password for the production user account.

## wordpress display order

If you want to change the order in which a payment method is displayed for a customer, click on the icon opposite the required payment method and, while holding the left mouse button, move this payment method to the required position.

## **cs-cart compatibility**

The payment plugin was tested for compatibility with CS-Cart version 4.6.x.

## **terms in the cs-cart interface**

In the CS-Cart interface, modules that extend the site's functionality are called «add-ons». In this description, the term «plugin» is used.

## **plugin settings only**

The description contains only those settings that relate directly to the payment plugin. The rest of the settings are related to the content management system (CMS).

## **terms in the Drupal interface**

In the Drupal interface, add-ons are called «modules». The payment plugin falls into the same category, however the term «plugin» is used in this description.

## **drupal description based on example**

A combined distribution [Commerce Kickstart](#) version 7.x-2.45 was used for the purposes of this description. This distribution combines Drupal version 7.54 and Drupal Commerce version 7.x-1.13.

## **opencart compatibility**

The plugin was tested on OpenCart/ocStore version 2.3.0.2.

## **the location of the log files**

The event log files are located here: <site folder>/system/storage/logs/

## **prestashop compatibility**

The instructions for installing and configuring the plugin for PrestaShop version 1.7.x are given as example.

## **netcat compatibility**

The plugin is compatible with Netcat versions 5.9.x and higher and has been tested on E-commerce editions.

### **avoid error in personal account when testing REST**

This situation can be avoided in one of the following ways:

- use the Incognito mode for REST requests, and work in your personal account in the browser window as usual;
- use different browsers to test REST requests and work in your personal account.

### **wsdl connection URLs**

Use the following URL to connect to the Test Service (WSDL):

<https://3dsec.sberbank.ru/payment/webservices/merchant-ws?wsdl>.

### **payment reversal conditions**

- With one-phase payments, payment reversal is possible for orders in the state **Deposited**, but only before the close of the day, that is, until 24:00 (Moscow time) on the day when the payment was made.
- In case of two-phase payments, a payment reversal can be processed for orders in the **Approved** status.

### **refund conditions**

Refunds are possible for payments in **Deposited** status.

### **passing clientId in payment by binding**

When `clientId` parameter is passed initially, the client's identifier is registered in the payment gateway and the data of their card is saved (except for the CVC code). On subsequent passes of this parameter, the client is redirected to a page where they can choose from the following options:

- use the previously saved data of the payment card – in this case, the client does not need to re-enter the data of the payment card for payment;
- enter the data of the new payment card – in this case, the data of the new payment card will be additionally linked to the identifier of this client.

### **what is a callback**

Callback notifications are not email notifications or telephone notifications. These are notifications received through the programming interface.

## **contact support to enable callback**

To enable the ability to receive checksum notifications and get a private key, contact technical support.

## **self-signed certificates cannot be used**

Self-signed certificates are not allowed. The certificate must be signed by a trusted certification authority (see above).

## **basic callback parameters**

The table shows the basic parameters. A number of additional parameters that will be passed in notifications can be specified in personal account.

## **parameters must be sorted alphabetically**

Notably, the «parameter\_name;parameter\_value» pairs must be sorted in direct alphabetical order by parameter name.

## **checksum check value**

If the checksums match, this notification is authentic and was sent by the payment gateway. Otherwise, it is likely that the attacker is trying to pass off their notification as a payment gateway notification.

## **use Latin characters**

Use Latin characters when filling out the form.

## **need an apple developer account**

To be able to complete this procedure, you need to have an Apple Developer account.

## **apple pay id creation rules**

The identifier should begin with the word **merchant**. The address of your live site should be specified in the reverse order. For example, for the site **sale.test.ru** the identifier will look as **merchant.ru.test.sale**.



## **checklist for testing apple pay**

Below there is a short list of actions that will allow you to test Apple Pay. Further details are available [in documentation](#) on the Apple website.

## **how to create a test account with apple**

For more information about creating a test account, see [Apple documentation](#).

## **create a new apple account**

If you mistakenly use a test account on a test device to log in to a production environment (such as iTunes) instead of logging in to a test environment, this test account will become invalid and cannot be used again. In this case create a new test account with a new email address.

## **samsung only for payment via mobile app**

Steps 1-3 are necessary to enable payment via the mobile app.

## **it is not necessary to wait for the completion of the report**

You can start generating a new report without waiting for the completion of the previous one.

## **number of generated reports**

Number in brackets after the caption **Reports** means how many reports are currently being generated.

## **fiscalization settings are set by support**

Fiscalization settings are set by technical support upon request to [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru). At this stage, you do not need to configure the fiscalization parameters yourself.

## **information about actions in atol personal account**

Detailed information about actions in the ATOL personal account is given on [ATOL website](#) (external link).

## **test data for connection to OFD**

Do not specify actual data to connect to the fiscal data operator in the test personal account! If you do so, test payments will be registered with the fiscal data operator as if they were real. After entering the data into the live personal account – check that they are valid and saved correctly.

## **passing empty compositecompletionamount**

If zero is passed in this parameter, completion will occur for the entire pre-authorized amount.

## **required parameters in getorderstatusextended**

Either orderId or orderNumber must be present in the request. If both parameters are present in the request, orderId has priority.

## **required parameters in getorderstatusextended ws**

The request must contain either orderId or merchantordernumber. If both parameters are present in the request, orderId has priority.

## **post only**

Only POST is supported.

## **the response depends on the OFD**

Some parameters to be returned may depend on the Fiscal Data Operator used.

## **use standard requests**

Use the standard requests to the payment gateway for the operations of reversal, refund and payment completion.

## **you need to pass the sbrf\_spasibo parameter**

If the merchant is connected to the «Spasibo from Sberbank» affiliate program, and bonus points are sent in the registration request, then the parameter should be passed as an additional parameter sbrf\_spasibo:amount\_bonus indicating the amount of bonus points to be deducted, as in the example below.

```
jsonParams={«sbrf_spasibo:amount_bonus»:«100»}
```

In this case, the amount in money will be deducted from the amount parameter.

### **you need to pass the sbrf\_spasibo parameter (ws)**

If the merchant is connected to the «Spasibo from Sberbank» affiliate program, and bonus points are sent in the registration request, then the parameter should be passed as an additional parameter `sbrf_spasibo:amount_bonus` indicating the amount of bonus points to be deducted, as in the example below.

```
<params name=«sbrf_spasibo:amount_bonus» value=«1030»/>
```

In this case, the amount in money will be deducted from the amount parameter.

### **reversals and refunds cannot be made**

Orders that initiate recurring payments cannot be reversed and refunded, as in these cases there is no real debiting.

### **passing an empty amount**

If you do not specify the amount parameter, completion will take place for the entire pre-authorized amount.

### **quantity and positionid are sufficient**

For error-free processing of the request it is sufficient to pass the quantity and positionId parameters.

### **passing empty depositamount**

If you do not specify the depositAmount parameter, completion will take place for the entire pre-authorized amount.

### **authcode field is obsolete**

The order status must be determined by the orderStatus parameter value. The authCode field is obsolete.

### **getreceiptstatus is not supported in ws**

Currently, requesting information about the status of a check in the WS interface is not supported.

## **download virtuemart**

From the VirtueMart developer site you can [download](#) the Joomla! distribution with pre-installed add-on VirtueMart. The names of such combined archives end with **Stable-Full\_Package.zip**. When doing this, choose a combination of Joomla! and VirtueMart, compatible with the payment plugin.

## **joomla compatibility**

The plugins are only guaranteed to work with the following combinations of Joomla! and VirtueMart add-ons. If you use other combinations of versions (for example, after updating Joomla!) The plugin may not work correctly.

## **compliance with 54-fz**

Starting from this version, the plugin allows you to transfer a shopping cart, which provides the possibility of detailed fiscalization in accordance with [54-FZ](#).

## **on the example of the next version of joomla**

Installing and configuring the plugin is presented on Joomla! version 3.6.5 and VirtueMart version 3.2.0.

## **android pay not supported**

The Android Pay payment method is currently not supported by Google. This payment method will be replaced by Pay with Google (it is currently temporarily unavailable for integration, documentation is under development).

## **section in development**

The section is currently under construction.

## **itemamount is rounded to the second decimal place**

When calculating  $\text{itemAmount} = \text{itemPrice} * \text{quantity}$  parameter, the result is rounded up to the second digit after the decimal point. For example, if the calculation result is 100.255, then final value will be 100.26.

## **orderid ordernumber or merchantlogin required**

To execute the request, one of the following parameters must be specified:

- 2;
- 2;
- 2.

### **access to payment method**

To be able to accept payment in this way, you must have the respective rights in the system. If in doubt, contact technical support.

### **only one-phase payments**

Payment in this way is possible only when using a one-phase payment scheme.

### **asymmetric cryptography is recommended**

For greater security, use of a method in which the checksum is generated using asymmetric cryptography is recommended.

### **the order of the parameters can be arbitrary**

The order of parameters in the notification can be undefined.

### **symmetric cryptography example**

Below are examples of creating a checksum using symmetric cryptography.

### **in order to familiarize**

The information on this portal is provided as an introduction to the functionality of the payment gateway. The necessary data for connecting to the live environment will be provided after testing in accordance with [connection checklist](#).

### **request only through rest**

Currently this request is available through REST interface only.

### **for samsung pay web you need a javascript script**

To be able to receive payment via Samsung Pay Web with a merchant-side payment page, you must install [JavaScript script](#). When receiving a response from the payment gateway, you need to call the

connect function of this script and pass in it certain parameters of the payment gateway response. For details, see the description of the corresponding [interaction schemes](#) and description of the [request](#).

### **https and tls are required**

Sites must use the HTTPS scheme and support the TLS 1.2 protocol.

### **safari required**

When making a payment from a web page, the buyer must be in the Safari browser.

### **programming skills required**

Programming skills are required to implement automatic check of payment completion.

### **don't share the private key with anyone**

Do not share or publish the private key to anyone. Signature verification must be implemented on the store's server. JavaScript implementation of signature verification on the front-end of the store is not allowed.

### **you need to have google merchant id**

To make payments from the website, your online store site must be registered with Google Pay.

### **for ofd, apply for connection**

Before registering a personal account of a cloud cash register provider [apply](#) to connect to Internet acquiring.

### **additional parameters**

If the merchant has fiscalization set up, when indicating email (the buyer's email address) and/or phone (the buyer's cell phone number) as additional parameters, these parameters are first used to send the fiscal receipt.

### **test and live mode for plugins**

- If the plugin is running in test mode, in the respective fields (**Login** and **Password**) you should

specify the data of the test service account.

- If the plugin works in the live mode, in the respective fields (**Login** and **Password**) you should specify the data of the live service account.

### **only if fiscalization is enabled**

This setting is applied only if you have configured fiscalization parameters – see details in [section on compliance with law 54-FZ](#).

### **fiscalization support for amiro**

Fiscalization is supported in plugins for Amiro.CMS versions 7.x and later (see [details on fiscalization](#)).

### **only for amiro 7 and higher**

The setting is available only in plugins for Amiro.CMS versions 7.x and later.

### **hold ctrl to select multiple items**

If you want to select multiple items, hold down the CTRL key on your keyboard before checking the items in the list.

### **plugin supports fiscalization**

The plugin supports fiscalization in accordance with the law [54-FZ](#).

### **features verify**

- Even if the payment amount is to be passed in the request, it will not be debited from the account.
- After the order has been successfully registered, it is put to the REVERSED (cancelled) status.

### **integration is not needed under the following conditions**

Integration with a payment gateway is not required if you use:

- [e-mail invoicing service](#);
- [payment links generator](#);
- work using [Yoom.Shop](#) (external ink);
- work using the following CMS:
  - [1C-Bitrix](#);
  - [WordPress](#);

- [Joomla!](#);
- use the transfer of goods by default from the fiscalization settings of the Sberbank personal account.

## **checking fiscalization on the live server**

After making all the settings, we ask you to carefully check the correct filling of all fields in relation to the fiscalization parameters in the personal account of the bank, because the correctness of the transfer of fiscal data to the OFD and tax depends on this. We also recommend checking «in action» when sending a fiscal receipt.

In case of any difficulties or questions, you can send them to the address of the technical support of the bank [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru) and Business.Ru [support@mail.business.ru](mailto:support@mail.business.ru).

## **google pay developer info**

Information for developers is available in [Google documentation](#).

### **information for Google Pay Web developers**

- Information for developers is available in [Google documentation](#).
- The payment page must be designed in accordance with [brand guidelines](#).

### **information for Google Pay In-App developers**

- Information for developers is available in [Google documentation](#).
- The application must be designed in accordance with [brand guidelines](#).

## **google tos**

By accepting payments via Google Pay, you agree to [Google terms of service](#).

## **passing the shopping cart is optional**

If the cart and fiscalization data are not included in the request, then the default values specified in the personal account are passed to the Fiscal Data Operator (see User Manual for personal account).

## **rest request type**

Using the REST protocol, you should send requests with the type `application/x-www-form-urlencoded`, rather than `multipart/form-data`.



## **not enough rights to submit a payment form**

If the section **Invoice to email** the following message is displayed: “You do not have sufficient rights to send the payment form”, please contact the technical support service support@ecom.sberbank.ru with a request to activate the invoicing option. Make sure to include the following information in your request:

- operator login (used to enter your personal account).
- environment (test/production) for which changes need to be made

## **QR code for payment link**

The link can also be encoded as a QR code.

## **closing Merchant Portal session**

If there are no user actions for **10+ minutes**, the session is automatically closed.

## **exporting charts from Merchant Portal**



A click on  saves the chart image.

## **you can select several statuses**

You can set several statuses at the same time by pressing CTRL and selecting them with a left-click.

## **short payment term**

If you set the invoice validity period too short, the client will not be able to use the same invoice to pay for the order after some time.

## **use of two-phase payment**

When using a two-phase payment, it is recommended to debit the amount no later than 7 calendar days (for some industries, such as the hotel business, car rental, etc., the period of possible holding may be increased).

## **change password for production**

Passwords provided for both logins in production environment must be changed in the personal

account (available [here](#)).

### **make sure to change password for all users**

Password for each login must be changed.

### **two-phase payment**

When using **two-phase** payment, it is recommended to debit the amount no later than 7 calendar days (for some industries, such as the hotel business, car rental, etc., the period of possible holding may be increased).

### **plugins support 54-fz**

#### **Fiscalization - 54-FZ**

Most of the plugins presented support fiscalization. To clarify whether the plugin you are interested in supports the fiscalization functionality, you can visit our technical support address – [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru).

### **support for plugins**

We are ready to develop plugin support for new CMS. To do this, you need to contact the technical support service – [support@ecom.sberbank.ru](mailto:support@ecom.sberbank.ru).

### **payment phases in Online Lending**

For online lending, only one-phase payment is applied.

### **only one-phase payment**

When using the Online Lending functionality, only the one-phase payment is available.

### **token authentication not possible**

When using the Online Lending functionality, token authentication is not possible. Login and password should be used for authentication.

### **only through rest**

Online Lending functionality is available through the REST interface only.

### **itemamount gift**

If all the items in the request are delivered without payment (a gift to customer), the value 0 should be passed.

### **itemprice gift**

If the item is delivered without payment (a gift to customer), the value 0 should be passed.

### **can be changed in refund**

When requesting a refund, the value of this parameter may differ from that given during the registration of the order.

### **passing additionalOfdParams**

The transfer of this block is possible only when using the following OFDs:

- ATOL;
- Business.Ru;
- Evotor.

### **ffd 105 and higher**

The settings are effective only if the format of fiscal documents of version 1.05 and higher is selected.

### **FFD version must match**

The format of the version must match the format selected in the personal account of the bank and in the account of the fiscalization service.

### **add json header**

It is necessary to add to the request a header with the definition of the type of contents – «Content-Type: application/json» to ensure that the request is processed correctly.

### **assumes joomshopping is installed**

This tutorial assumes that your Joomla! the JoomShopping component is already installed.

### **assumes virtuemart is installed**

This tutorial assumes that your Joomla! the VirtueMart component is already installed.

### **simpla version**

The plugin was tested for Simpla version 3.2.x.

### **google pay registration**

These steps are necessary if you want to receive encrypted payment actions directly from Google Pay and redirect them to a payment gateway. Namely:

- online payment, if the payment page is located on the merchant's side;
- payment from the application.

### **google pay info**

For more information see [Integration Checklist](#) (external link).

### **google pay create account**

It is recommended to use an existing email address on your domain, for this you need to select the option **Use current email address**.

### **samsung pay registration**

These actions are necessary to enable payment via the mobile app. You do not need to take these steps to pay through the site.

### **shop-script version**

Shop-Script version must be 7.x or higher.

### **not for ofd.ru**

Not applicable for **OFD.ru**.

## **magento compatibility**

Magento plugins for versions 2.2.x and 2.3.x are incompatible. When upgrading Magento version from 2.2.x to 2.3.x, you must first uninstall the plugin for version 2.2.x and then install the plugin for version 2.3.x.

## **php folder**

The beginning of the command depends on the folder where PHP is installed. In particular, the PHP version can be specified without a hyphen: php5.6, php7.1, php7.2, etc.

If in doubt, contact your hosting provider.

## **may not have enough memory**

On shared hosting, the last command may not be executed due to the memory limit set by the hosting provider. In this case, contact your hosting provider.

## **uninstalling magento**

In code folder, as well as in the folders of a higher level, (app) there may be data not related to the plugin, so delete only the Rbs folder and its contents.

## **6 and 8 are not used**

Types 6 (gambling winnings) and 8 (lottery winnings) are not used.

## **initpro start of integration**

At this step, you can right away configure [automatic addition of your cash registers](#) to the personal account of the OFD.

## **initpro requires an EDS**

To register a cash desk through the FTS website, you will need an EDS. If you don't have it, you can [order it from us](#) or submit a paper application for registering the cash register with the tax service.

## **initpro free code**

You can find the free OFD activation code in [cash register profile](#) in the personal account of Initpro.

## **no integration setup required**

You don't need integration setup if:

- you are using [email invoicing service](#);
- you are using [payment link generator](#);
- you are working with the following CMSs: [1C-Bitrix](#), [WordPress](#), [Joomla!](#);
- use the default product transfer from [fiscalization settings](#) of Merchant Portal

Sberbank.

## **closeOfdReceipt**

- A closing receipt can be sent multiple times for one order. There is no limit on the amount, that is, the amount indicated in the closing receipt may be less or greater than the order amount.
- An exception is made for orders for which a refund has been made. For orders for which a refund was made, the closing of the check can only be done if the total of all refunds is less than the confirmed order amount (i.e., there was no full refund).

## **closing if refund**

Closing a check can only be done if a partial, not full refund has been made for the order.

## **amount info**

If you specify amount=0 in the request, a refund is made for the entire amount of the order.

## **amount credit**

Before summing up all line items, the product of quantity and price for each good item is rounded to an integer. If the value is followed by 5 or greater after the decimal point, it is rounded up. Below there are examples of rounding.

1. If quantity = 0.111, and price = 5500, then the result is 611 (610.5 rounded up).
2. If quantity = 1.455, and price = 6900, then the result is 10040 (10039.5 rounded up).
3. If quantity = 1.211, and price = 6988, then the result is 8462 (8462.468 rounded down).

Therefore, the «amount» parameter value must be equal to the sum of the line items rounded.

## **credit only rest**

Online Lending functionality is available through the REST interface only.

### **items credit warning**

Do not use «')» character combination inside this section, or it will cause an error on the gateway side.

### **itemprice info**

- itemPrice cannot be less than 0.
- itemPrice may be either string or integer.

### **getreceiptstatus warning**

To execute the request, the order number or its identifier in the fiscalizator must be specified.

### **customerdetails info credit**

A parameter included in a block and indicated as mandatory is only mandatory within that block. If the block itself is not mandatory and is not passed, its included parameters should not be passed.

### **the ability to make two-phase payments**

Two-phase payments functionality is available if specifically requested by Merchant from the bank.

### **preauth parameter**

This parameter can be passed either as a boolean value (without quotation marks) or as a string (with quotation marks).

### **email ОФД**

Most OFDs support delivery of receipt only to one email address. Some OFDs allow receipt delivery to multiple emails. In this case, the emails should be separated by commas without spaces. Contact your OFD for availability of receipt delivery to multiple emails.

### **positionid warning**

The registered cart must contain an item with this positionId value, otherwise the order will fail.

### **callback delimiter**

Use semicolon (»;) without a space as the separator. The string should end with semicolon (»;) as

well.

### **sending a notification is considered unsuccessful**

If a response other than the 200 OK HTTP code is returned to the payment gateway, the notification sending is considered to be unsuccessful. In this case, the payment gateway repeats sending the notification (the first attempt is made after 30 seconds, and the next with an interval of 10 minutes) until one of the following conditions is met:

- the payment gateway receives the 200 OK HTTP code in response to the callback notification

or

- there are six unsuccessful attempts to inform in sequence.

When either of the above conditions is met, attempts to send a callback notification about the operation are terminated.

### **REST authorization**

The transfer of authorization data (username/password) can be transferred both in the request body and in the header.

### **standard refund and reversal requests**

Refund and reversal requests are standard payment gateway requests.

### **token from tech support**

The token value can be obtained from the technical support staff.

### **autocompleteondate**

To enable this functionality, please contact technical support.

### **itemdetails warning**

The field size limit is 1024 bytes.

### **name (items)**

Use



- to pass \ use \«- to pass »

## **phone note**

If the number is passed in a separate parameter and in additional parameters, the number specified in this phone parameter will be used.

## **customization of the payment page**

Currently, minimal customization of the payment page is available, only the following elements can be changed:

- logo;
- footer.

The requirements for these elements are presented in the instructions for working with your personal account.

## **recurrent payment auto payments**

For recurring payments via SberPay, only one-phase payment can be used. At the same time, for ordinary bindings, a two-phase payment is possible.

## **data types**

In this document, the following data type conventions are used when describing request and response parameters:

- A<n> - a sequence of Latin letters of length <n>;
- A..<n> - a sequence of Latin letters with a length not exceeding <n>;
- N<n> - a sequence of digits of length <n>;
- N..<n> - a sequence of digits with a length not exceeding <n>;
- AN<n> - a sequence of Latin letters and numbers of fixed length <n>;
- AN.. <n> - a sequence of Latin letters and numbers with a length not exceeding <n>;
- ANS<n> - a sequence of Latin letters, numbers and characters of fixed length <n>
- ANS.. <n> - a sequence of Latin letters, numbers and characters with a length not exceeding <n>;
- UTC - date and time, in this case: the date must be passed without specifying the time zone, Moscow time, for the SOAP protocol, the standard encoding xs: dateTime is used.

## **data types 2**

In this document, the following data type conventions are used when describing request and response parameters:

- string – string;
- array – data array;
- object – data object;
- data – date;
- boolean – boolean (logical) data type;
- int. (integer) – integer data type;
- amount – a numeric data type (amount).

## ATOL phone

When using ATOL fiscalization, the phone parameter should be sent in the following format:

The phone number must be passed together with the country code without spaces and additional characters except for the quotation marks (the number «371 2 1234567» must be passed as «+37121234567»). If the phone number belongs to Russia (prefix «+7»), then the value can be passed without a prefix (the number «+79251234567» can be sent as «9251234567»). The maximum string length is 64 characters. One of the fields must be filled in the request: email or phone.

## reserved names in params, jsonParams, additionalParameters (Spasibo)

It is forbidden to pass reserved names in the parameter (if they are passed, the order may be rejected):

- sbrf\_spasibo:amount\_bonus
- sbrf\_sbermiles:amount\_bonus
- loyaltyId

## possible values of measure

Possible values of **measure**

Value	Description
0	Used for payment subjects that can be sold individually or in units (as well as in the event that the payment subject is a product subject to mandatory labeling with an identification tool (which can be known from mark_code being passed))
10	Gram
11	Kilogram
12	Ton
20	Centimeter
21	Decimeter
22	Meter
30	Square centimeter
31	Square decimeter
32	Square meter
40	Milliliter
41	Liter

Value	Description
42	Cubic meter
50	Kilowatt hour
51	Gigacalorie
70	Day (24h)
71	Hour
72	Minute
73	Second
80	Kilobyte
81	Megabyte
82	Gigabyte
83	Terabyte
255	Applicable when using other units of measurement

## markQuantity

Parameters of **markQuantity** tag:

*A parameter included in a tag and indicated as mandatory is only mandatory within that tag. If the tag itself is not mandatory and is not passed, its included parameters should not be passed.*

Name	Type	Mandatory	Description
numerator	N..12	yes	The numerator of the fractional part of the payment subject.
denominator	N..12	yes	The denominator of the fractional part of the payment subject.

The table below shows how the merchant and the payment gateway interact when a payment is refunded.

	<p>The merchant makes a refund request to the buyer:</p> <ul style="list-style-type: none"> <li>• full refund: <ul style="list-style-type: none"> <li>◦ <a href="#">REST interface (refund.do)</a>;</li> <li>◦ <b>1</b> <a href="#">WSDL interface (refundOrder)</a>;</li> </ul> </li> <li>• partial refund: <ul style="list-style-type: none"> <li>◦ <a href="#">REST interface (refund.do)</a>;</li> <li>◦ <a href="#">WSDL interface (refundOrder)</a>.</li> </ul> </li> </ul>
<p><b>2</b></p>	<p>After receiving a successful response from the Payment Gateway, the Merchant makes a request for order status:</p> <ul style="list-style-type: none"> <li>• <a href="#">REST interface (getOrderStatusExtended.do)</a>;</li> <li>• <a href="#">WSDL interface (getOrderStatusExtended)</a>.</li> </ul>

There is an option to set up [callback notifications](#) about order status instead of sending the getOrderStatusExtended request. These notifications will be sent to you automatically after the order status changes.

## Operations with bindings

Request name	Links to request specifications
Request for a binding deactivation	<ul style="list-style-type: none"><li>• <a href="#">REST interface</a>;</li><li>• <a href="#">WSDL interface</a>.</li></ul>
Request for activation of an binding that has been previously deactivated	<ul style="list-style-type: none"><li>• <a href="#">REST interface</a>;</li><li>• <a href="#">WSDL interface</a>.</li></ul>
Request for the list of bindings by customer ID	<ul style="list-style-type: none"><li>• <a href="#">REST interface</a>;</li><li>• <a href="#">WSDL interface</a>.</li></ul>
Request for the list of bindings of a certain bank card	<ul style="list-style-type: none"><li>• <a href="#">REST interface</a>;</li><li>• <a href="#">WSDL interface</a>.</li></ul>
Request for changing the validity period of a binding	<ul style="list-style-type: none"><li>• <a href="#">REST interface</a>;</li><li>• <a href="#">WSDL interface</a>.</li></ul>

Was this page helpful?

(1) (1)