

SberPay payment widget for placing payment on the merchant's page

The module is designed to be placed on the merchant's pages in order to organize payment for services by using the Sberbank Online service without leaving the store page.

Connection

On the merchant's website

To enable the functionality on **their own website** the merchant should place the following code on their page in the `<head></head>` block:

```
<head>
  ...
  <script
src="https://securepayments.sberbank.ru/payment/modules/sbol-pay/sbol-pay.js
"></script>
  ...
</head>
```

On the payment page

To enable the functionality on **the payment page** the merchant should place the following code on their page in the `<head></head>` block:

```
<head>
  ...
  <script src="../../modules/sbol-pay/sbol-pay.js"></script>
  ...
</head>
```

This applies to payment pages located on one of the URLs:

- UAT - <https://3dsec.sberbank.ru/...>
- PROD - <https://securepayments.sberbank.ru/...>

and which do not have logins: `rbs`, `sbersafe`, `sbersafe_id`, `sbersafe_cardholder`, `sbersafe_sberid` (since this module is already there).

Script address for test and production environments

- **Testbed (UAT)** – <https://3dsec.sberbank.ru/payment/modules/sbol-pay/sbol-pay.js>
- **Production Environment (PROD)** – <https://securepayments.sberbank.ru/payment/modules/sbol-pay/sbol-pay.js>

Initializing the script

It is recommended to initialize the script at the end of the document, before closing the <body> tag, by placing a block similar to the one below:

```
<script>
  document.addEventListener("DOMContentLoaded", function() {
    var sbolWidget = new window.SbolPay({
      // DOM element selector on the page where the payment widget will be
      added
      selector: '#sbol-pay-container',
      // The address to which you want to redirect the user in case of
      successful payment
      returnUrl: 'https://example-shop.com/success'
      // URL to which the user is to be redirected in case of an uncessful
      payment
      failUrl: 'https://example-shop.com/fail'
      // Individual public key of the merchant
      token: 'j5d0bk0h860v407jg00806ab0p',
      // Payment amount in kopecks
      amount: '123400',
      // Name of the store to display in the widget window
      shopName: 'Promtovary LLC',
      // the order number in the store system
      orderNumber: '123456',
    });
  });
</script>
```

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

Available widget initialization parameters

Below is a complete list of parameters for initializing the script. Parameters marked as required must be specified during initialization and do not have a default value.

Parameter	Mandatory	Default value	Description
-----------	-----------	---------------	-------------

Parameter	Mandatory	Default value	Description
selector	Yes	-	DOM element selector where the payment widget will be placed.
token	Yes	-	The merchant's public key. To get the public key, contact technical support.
amount	Yes	-	The order amount. Indicated in minor units (in kopecks). For example, to write the value of 10 rubles. you need to specify the value «1000» (in string format)
returnUrl	Yes	-	The URL to which the user is to be redirected in case of a successful payment (and also in case of a failed payment where the failUrl parameter is not passed). The address must be specified in full including the protocol used (for example, https://test.ru instead of test.ru).
failUrl	No	-	The address to which the user is to be redirected in case of a failed payment. The address must be specified in full including the protocol used (for example, https://test.ru instead of test.ru). The parameter is optional. In this case, an unsuccessful payment will lead to a redirect to returnUrl, in the same way it happens when the payment is successful.
shopName	No	-	The name of the store to display in the widget window.
clientId	No	-	Identifier of the Customer in the Store system.
orderNumber	Yes	-	Order number in the store system. This is optional only if the automatic generation of the order number is enabled on the gateway (for this, contact technical support).
description	No	-	Order description.
jsonParams	No	-	Additional order parameters in object format. 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

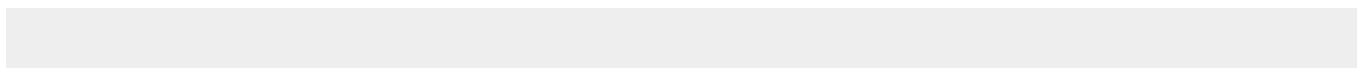
Parameter	Mandatory	Default value	Description
cartItems	No	-	An array of objects for describing the items in the cart. Example: <pre>[{ // Description of the line item name: 'Ball pen', // Object describing the total number of line items quantity: { // Number of line items value: 1, // Measure of the quantity of a line item measure: "pcs" }, // The sum of the cost of all line items of the specified type itemAmount: 23400, // Number (identifier) of a line item in the store system itemCode: 'code_1' }, { name: 'Cheese. Russian', quantity: { value: 1.4, measure: "kg" }, itemAmount: 100000, itemCode: 'code_2' }]</pre>
onlyButton	No	false	An indication that the widget will be displayed with one button without a container with a description.
flatButton	No	-	A sign that makes the style of the widget call button flat (instead of a gradient). Example: 
rowView	No	false	Display the widget as a linear view. Example: 
rowViewMinWidth	No	400	The width of the page window at which the rowView style is applied (valid only if the value is set to «rowView: true»)
phone	No	-	Phone of the customer making the payment. If the phone is known in advance, you can transfer it to the option. In this case, the widget will not ask the user for a phone number, but will immediately send a confirmation of payment to the specified phone number. The number must be specified in the format: «9001234567».

Parameter	Mandatory	Default value	Description
noStyle	No	false	Do not add styles defining the widget appearance. (The website administrator might need to use custom styles for the widget).

Methods

Name	Description
destroy	Method to remove the triggered button. <pre><script> document.addEventListener("DOMContentLoaded", function() { var sbolWidget = new window.SbolPay({ // Options }); function funcExample() { sbolWidget.destroy(); } }); </script></pre>
.on	Method for keeping track of the state of the widget. <pre>// Initialization as usual const sbolPay = new window.SbolPay(...); // handling widget status changes sbolPay.on('CHANGE_WIDGET_STATUS', function (event) { console.log('new status:', event.status, 'old status:', event.previousStatus); });</pre> Possible states: • IDLE - modal widget window is closed • IN_PROGRESS - modal widget window is open Example: <pre>// Initialization as usual const sbolPay = new window.SbolPay(...); // handling widget status changes sbolPay.on('CHANGE_WIDGET_STATUS', function(event) { if (event.status === 'IDLE') { // The user closed the modal window } else if (event.status === 'IN_PROGRESS') { // User opened modal window } });</pre>

An example of a page code with a connected module:



```

<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SBOL Example</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.c
ss">
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></sc
ript>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery.inputmask/5.0.3/jquery.in
putmask.js"></script>

  <script
src="https://3dsec.sberbank.ru/payment/modules/sbol-pay/sbol-pay.js"></scrip
t>
  <style type="text/css">
    body {
      padding: 20 px;
    }
    h1 {
      font-size: 18 px;
      margin-top: 40 px;
    }
  </style>
</head>
<body>
  <h1>Payment by entering a phone number in the module</h1>
  <div id="container"></div>

  <h1>Payment by entering the number on the merchant's page</h1>
  <div class="row">
    <div class="col-3">
      <div class="form-group">
        <label for="exampleInputEmail1">Phone number</label>
        <input type="text" id="phone" class="form-control">
      </div>
    </div>
  </div>
  <div id="container_with_phone"></div>
  <script>
    // Wait for all DOM elements to load
    document.addEventListener("DOMContentLoaded", function() {
      $('#phone').inputmask({"mask": "+7(999) 999-9999"});

      // Initialization of the script for payment by entering the phone
      number inside the module
      var sbolWidget = new window.SbolPay({

```

```

        // DOM element selector on the page where the payment widget will be
added
        selector: '#container',
        // Individual public key of the merchant
        token: 'j5djbk3h86qv4k7jg40826abbp',
        // Payment amount in kopecks
        amount: '123400',
        // Name of the store to display in the widget window
        shopName: 'Promtovary LLC',
        returnUrl: 'http://yandex.ru'
    });

    // Initialization of the script for payment by entering the phone
number on the merchant's website
    // Reinitialized when the phone number changes and if it is completely
full
    $('#phone').on('input', function(e) {
        var rawPhone = $(this).val().replace(/\D/g,'').slice(1);

        // Check for the length of the number. Supported format 9123456789
        if (rawPhone.length === 10) {
            var sbolWidgetPhone = new window.SbolPay({
                // DOM element selector on the page where the payment widget
will be added
                selector: '#container_with_phone',
                // Individual public key of the merchant
                token: 'j5djbk3h86qv4k7jg40826abbp',
                // Payment amount in kopecks
                amount: '123400',
                phone: rawPhone,
                // Name of the store to display in the widget window
                shopName: 'Promtovary LLC',
                returnUrl: 'http://yandex.ru'
            });
        }
    })
});
</script>
</body>
</html>

```