

Payment button

A simple way to ensure that payments are accepted on any website is to place a button next to the description of a product or service that will open a payment form in a modal window (iframe). An example of using the payment button is available at [this page](#).

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

Details

Preliminary actions

Before you start preparing your site, you should get a special token from technical support: an alphanumeric value that should be used in the site code to create a payment button. On this site the following token will be used as example: YRF3C5RFICWISEWFR6GJ.

Changes to the `<head>` tag of the store page

To embed a checkout button on a store page, you must modify the content of the `<head>` tag of the page that will host the checkout button.

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»>`.

In the `<head>` tag of the page add:

- path to JavaScript script (`ipay.js`) hosted on the payment gateway server;
- merchant token obtained from technical support.

```
<head>
  <title>Test Page</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width">
  <script
src="https://3dsec.sberbank.ru/payment/docsite/assets/js/ipay.js"></script>
  <script>
    var ipay = new IPAY({api_token: 'YRF3C5RFICWISEWFR6GJ'});
  </script>
</head>
```

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

Changes in the body of the store page

In the body of the page, you need to add a link to the JavaScript script (ipay-modal.js), in which you can specify the following parameters.

Name	Description	Mandatory	Type
amount	Payment amount. Allows you to specify the amount in rubles with kopecks, while a period is used as a separator, for example: 499.99 (four hundred and ninety-nine rubles, ninety-nine kopecks). If you do not specify the payment amount, the buyer will be able to enter an arbitrary amount on their own.	No	N..12
currency	Payment currency. ISO 4217 alphabetic or numeric currency codes can be used, for example RUB.	Yes	N3
order_number	Order number. If you do not specify a parameter value, the order number will be generated automatically by the payment gateway.	No	ANS..32
description	Order description. Usually this is the name of the product or goods.	No	ANS..512

Code examples are given below:

- [the exact price is specified](#);
- [the exact price is not specified](#).

The exact price is specified

```
<a onclick="ipayCheckout({
  amount:499.99,
  currency:'RUB',
  order_number:'',
  description: 'A. S. Pushkin. Selected works (deluxe edition) ',
  function(order) { showSuccessfulPurchase(order) },
  function(order) { showFailurefulPurchase(order) })"
  class="btn btn-xs btn-outline btn-primary">Buy
</a>
```

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

An example of the implementation of a payment button with an indication of the exact price.

The exact price is not specified

```
<a onclick="ipayCheckout({
  currency: 'RUB',
  order_number: '',
  description: 'N. V. Gogol. Evenings on a Farm Near Dikanka'},
  function(order) { showSuccessfulPurchase(order) },
  function(order) { showFailurefulPurchase(order) })"
  class="btn btn-xs btn-outline btn-primary">Buy
</a>
```

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

An example of the implementation of a payment button indicating an arbitrary price.

Checking the completion of payment

When using the payment button, interaction with the payment gateway occurs in an unprotected way – through elements embedded in the store's website. In this case, the buyer can change the properties of the payment, for example, the amount. Therefore, after completing the payment, it is important to check that the payment made meets the original requirements.

Manual verification of payment completion

No programming skills are required for manual payment verification. You can check the completion of the payment manually in two ways:

- check the parameters of the actual payment in [personal account](#);
- check the notification email received from the payment gateway upon payment.

Automatic check of payment completion

Automatic check of the completion of payment involves verification of the digital signature in [callback notification](#) on the store server.

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

After the client completes the payment, one of two functions will be called: `successCallback (order)` or `failureCallback (order)`, which you need to implement in order to perform the necessary payment-related actions (send the goods for delivery, etc.). When `successCallback` and `failureCallback` functions are called, a set of key payment data and a signature generated using the private key issued during registration are passed as a parameter (`secretToken`).

The signature is formed according to the following algorithm: the values of all attributes of the object `order` are concatenated into one line in the following order: `status`, `formattedAmount`, `currency`, `approvalCode`, `orderNumber`, `panMasked`, `refNum`, `paymentDate`, `formattedFeeAmount`. Then the private key is added to the end and the hash is calculated from the resulting string using the SHA-256 algorithm. The resulting value is passed in the `digest` property of `order` object.

The HMAC algorithm is used for authentication. The received data can be checked on resources, for example: <http://beautifytools.com/hmac-generator.php>

For example, for an `order` object like

```
{
  "approvalCode": "123456",
  "orderNumber": "310fe4-ec02-4d7f-af04-091f908bf0",
  "panMasked": "411111**1111",
  "refNum": "111111111111",
  "digest": "d55626a41f5713d3b46e4efa71301d2714e70bd6301f8dfd4bbfd00af668b498",
  "currency": "643",
  "paymentDate": "2016-04-28 20:58:24.25",
  "formattedAmount": "500.00",
  "formattedFeeAmount": "0.00",
  "status": "DEPOSITED"
}
```

`digest` field value was calculated by string

```
DEPOSITED500.00643123456310fe4-ec02-4d7f-
af04-091f908bf0411111**11111111111111112016-04-28
20:58:24.250.00j64h6ksafvm200ecpc2l1u4b9k;
```

where `j64h6ksafvm200ecpc2l1u4b9k` is an example of a private key.

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 can also check the fact of payment using [Payment gateway API](#) by making a `getOrderStatusExtended` request ([REST](#) or [WSDL](#) interface).

Was this page helpful?

(1) (1)