PAYEER

Connecting

To connect to Payeer:

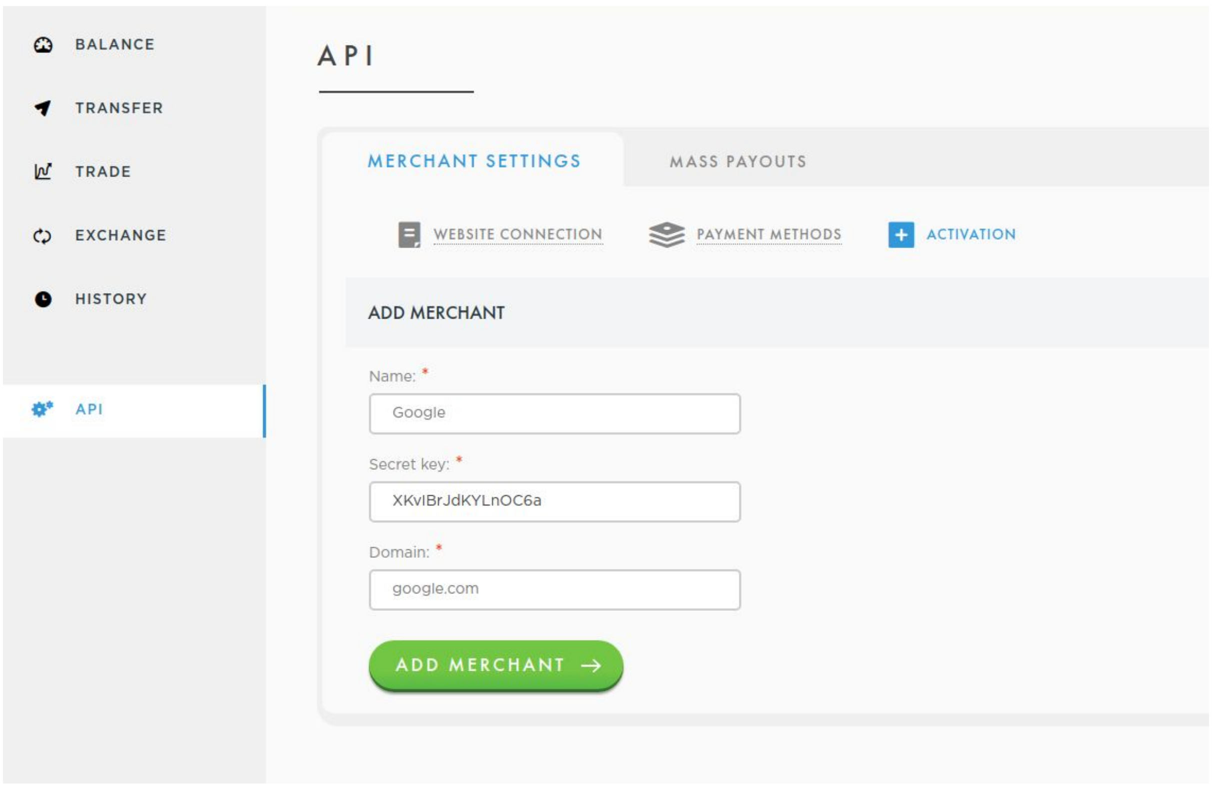
1. Register in the Payeer system

2. Activate the merchant in the “Merchant Settings” tab by going to the “Activation” tab and filling in the following fields:

Name: the name of your website, e.g. “Google”. This name will be displayed on your merchant list and the user’s list when paying an invoice.

Secret key: a string of characters for signing data transmitted in the payment initialization form and checking incoming data when processing a payment.

Domain : your website’s domain name, e.g. “google.com”. The domain name can only contain Latin letters, numbers, and hyphens. To convert national domain names, use any punycode converter such as punycoder.



Name: *
Google

Secret key: *
XKvIBrJdKYLnOC6a

Domain: *
google.com

3. Confirm your ownership of the website by placing a text file in the website's root directory.

4. Fill in the following fields:

Success URL: this is the address the customer will be redirected to once payment has been completed successfully.

Fail URL: this is the address the customer will be redirected to in the event of an error during the payment process or if payment is cancelled.

Status URL: the address of the payment processor. This page is where orders can be marked as paid or, for example, funds can be sent to the customer's account on your website.

Key for encrypting additional parameters: a secret key for encrypting additional fields, as well as dynamic success, fail, and status URL's.

5. Submit the merchant for moderation. Prior to moderation, only the owner of the merchant can perform payment as long as this person has already logged into Payeer.

Setting up the Merchant

In the "Settings" tab, you can:

1. Enter a new secret code (unfortunately, it is not possible to view the old code for security reasons, but you can always replace it with a new one).

2. Establish who will pay fees:
2.1. By default: the customer pays the fee for the selected payment method and the merchant pays a fee to Payeer (by default, 0.95%); the amount of the invoice minus 0.95% is added to the merchant's account.
2.2. Customer: the payment method's fee and Payeer's fee are both paid by the customer, and the full price of the invoice is added to the merchant's account
2.3. Merchant: the payment method's fee and Payeer's fee are both paid by the merchant, and the price of the invoice minus both fees is added to the merchant's account.

3. Enter a key for encrypting additional parameters if you are planning to transfer them to the payment form or you need dynamic interaction addresses (success, fail, or processor's address).

4. Enter the addresses of successful and failed payments and the payment processor.

5. Select a notification server (by default, the server is selected automatically when a notification is sent).

In the “Appearance” tab you can disable unneeded sections and payment methods or set only the currencies you need. Methods unavailable for direct payment of the invoice are grayed out, but your customer can always add funds to their internal Payeer account and then pay your invoice from that account. After clicking on these methods, the customer is provided with detailed instructions on how to return to paying your invoice.

In the “Website connection” tab you can download a ready-made module for your CMS and find examples for manual connection.

To test a signature that has been generated, you can use the tab of the same name in the merchant’s settings.

Payment Initialization Form

For switching to the payment page, create the following form:

```
<form method="post" action="https://payeer.com/merchant/">
	<input type="hidden" name="m_shop" value="12345">
	<input type="hidden" name="m_orderid" value="1">
	<input type="hidden" name="m_amount" value="1.00">
	<input type="hidden" name="m_curr" value="USD">
	<input type="hidden" name="m_desc" value="dGVzdA==">
	<input type="hidden" name="m_sign"
value="9F86D081884C7D659A2FEAA0C55AD015A3BF4F1B2B0B822CD15D6C15B0F0
0A08">
	<!--
	<input type="hidden" name="form[ps]" value="2609">
	<input type="hidden" name="form[curr[2609]]" value="USD">
	-->
	<!--
	<input type="hidden" name="m_params" value="">
	-->
	<input type="submit" name="m_process" value="send" />
</form>
```

Explanation of the Parameters of the Payment Initialization Form

Name	Key	Description
Merchant URL	***action***	The merchant's URL address
Merchant ID	***m_shop***	The merchant’s ID
Payment ID	***m_orderid***	In this field, the seller enters a purchase ID that matches their invoicing system. Using a unique number for each payment is recommended. The ID can be a line of any length up to 32 characters and can contain the characters “A-Z”, “_”, “0-9”, and “-”. **Example:** 12345
Payment amount	***m_amount***	The amount of the payment the seller wishes to receive from the purchaser. The amount must be greater than zero. There can only be two decimal places, and they must be separated from the whole number by a period. **Example:** 1.00
Payment currency	***m_curr***	The currency used in the payment **Potential currencies:** USD, EUR, RUB
Payment description	***m_desc***	A description of the product or service. Generated by the seller. This line is added to the payment designation. It is encoded by a base64 algorithm. **Example:** dGVzdA==

		PHP код: <?php echo base64_encode('test'); ?>
Electronic signature	***m_sign***	A security signature used to check the cohesiveness of the information obtained and directly identify the sender. **Example:** 9F86D081884C7D659A2FE AA0C55AD015A3BF4F1B2 B0B822CD15D6C15B0F00 A08
Payment system ID	*form[ps]*	The ID of the payment system for automatic selection. The ID list can be found in the **Appearance** tab **Example:** 2609
Payment system currency	*form[curr[psId]]*	Payment system currency **Example:** USD
Additional parameters	*m_params*	A JSON array of data of additional parameters encrypted using the Rijindael-256 algorithm and encoded using a base64 algorithm.
Encryption method	*m_cipher_method*	Encryption method of m_params. Available methods: ● AES-256-CBC ● AES-256-CBC-IV

Additional parameters

Name	Key	Description
Success URL	*success_url*	This is the address the customer will be redirected to once payment has been completed successfully
Fail URL	*fail_url*	This is the address the

		customer will be redirected to in the event of an error in the payment process or if payment is cancelled
Status URL	*status_url*	The address of the payment processor. This page is where orders can be marked as paid or, for example, funds can be sent to the customer’s account on your website
Additional fields	*reference*	An array of additional fields that must be returned to the payment processor
Domain of sub-merchant	*submerchant*	Domain of sub-merchant (only for aggregators)

Example of the Creation of a Payment Initialization Form

PHP

```
<?php
$m_shop = '12345'; // merchant ID
$m_orderid = '1'; // invoice number in the merchant's invoicing system
$m_amount = number_format(1, 2, '.', ''); // invoice amount with two decimal places
following a period
$m_curr = 'USD'; // invoice currency
$m_desc = base64_encode('Test'); // invoice description encoded using a base64
algorithm
$m_key = 'Your secret key';

// Forming an array for signature generation
$arHash = array(
	$m_shop,
	$m_orderid,
	$m_amount,
	$m_curr,
	$m_desc
);

/*
// Forming an array for additional parameters
$arParams = array(
	'success_url' => 'http://google.com/new_success_url',
	'fail_url' => 'http://google.com/new_fail_url',
	'status_url' => 'http://google.com/new_status_url',
		// Forming an array for additional fields
		'reference' => array(
				'var1' => '1',
				'var2' => '2',
				'var3' => '3',
				'var4' => '4',
				'var5' => '5',
		),

		//'submerchant' => 'mail.com',
);

// Forming a key for encryption
$key = md5('Key for encrypting additional parameters'.$m_orderid);

// Encrypting additional parameters
//$m_params = urlencode(base64_encode(mcrypt_encrypt(MCRYPT_RIJNDAEL_256,
$key, json_encode($arParams), MCRYPT_MODE_ECB)));

// Encrypting additional parameters using AES-256-CBC (for >= PHP 7)
$iv = substr(hash('sha256', $key), 0, 16);
$m_params = urlencode(base64_encode(openssl_encrypt(json_encode($arParams),
'AES-256-CBC', $key, OPENSSL_RAW_DATA, $iv)));

// Adding parameters to the signature-formation array
$arHash[] = $m_params;
*/

// Adding the secret key to the signature-formation array
$arHash[] = $m_key;

// Forming a signature
$sign = strtoupper(hash('sha256', implode(':', $arHash)));

$arGetParams = array(
		'm_shop' => $m_shop,
		'm_orderid' => $m_orderid,
		'm_amount' => $m_amount,
		'm_curr' => $m_curr,
		'm_desc' => $m_desc,
		'm_sign' => $sign,
		//'m_params' => $m_params,
		//'m_cipher_method' => 'AES-256-CBC-IV',
		//'form[ps]' => '2609',
		//'form[curr[2609]]' => 'USD',
);

$url = 'https://payeer.com/merchant/?'.http_build_query($arGetParams);
echo $url;
```

```
?>
<form method="post" action="https://payeer.com/merchant/">
	<input type="hidden" name="m_shop" value="<?=$m_shop?>">
	<input type="hidden" name="m_orderid" value="<?=$m_orderid?>">
	<input type="hidden" name="m_amount" value="<?=$m_amount?>">
	<input type="hidden" name="m_curr" value="<?=$m_curr?>">
	<input type="hidden" name="m_desc" value="<?=$m_desc?>">
	<input type="hidden" name="m_sign" value="<?=$sign?>">
	<?php /*
	<input type="hidden" name="form[ps]" value="2609">
	<input type="hidden" name="form[curr[2609]]" value="USD">
	*/ ?>
	<?php /*
	<input type="hidden" name="m_params" value="<?=$m_params?>">
	<input type="hidden" name="m_cipher_method" value="AES-256-CBC-IV">
	*/ ?>
	<input type="submit" name="m_process" value="send" />
</form>
```

Java

```
import java.util.*;
import java.lang.*;
import java.security.MessageDigest;
import java.util.Base64;


class Rextester
{
    public static void main(String args[])
    {
        String m_shop = "12345";
        String m_orderid = "1";
        String m_amount = "1.00";
        String m_curr = "USD";
        String m_desc = "Test invoice";
        String m_key = "Your secret key"";

        String m_desc64 = Base64.getEncoder().encodeToString(m_desc.getBytes());

        String m_sign = sha256(m_shop + ":" + m_orderid + ":" + m_amount + ":" + m_curr +
":" + m_desc64 + ":" + m_key);

        String url = "https://payeer.com/merchant/?m_shop=" + m_shop + "&m_orderid=" +
m_orderid + "&m_amount=" + m_amount + "&m_curr=" + m_curr + "&m_desc=" +
m_desc64 + "&m_sign=" + m_sign;

        System.out.println(url);
    }

    public static String sha256(String base)
    {
        try
        {
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hash = digest.digest(base.getBytes("UTF-8"));

            return bytesToHex(hash).toUpperCase();
        }
        catch(Exception ex)
        {
            throw new RuntimeException(ex);
        }
    }

    public static String bytesToHex(byte[] bytes)
    {
        StringBuffer result = new StringBuffer();
        for (byte b : bytes) result.append(Integer.toString((b & 0xff) + 0x100,
16).substring(1));
        return result.toString();
    }
}
```

Forming a Digital Signature

A digital signature is a line of characters made up of values of transmitted variables, with the addition of a secret key at the end, divided by the character “:” (colon) and hashed using an SHA-256 algorithm. All letters are capitalized.

Sample Signature Formation

PHP

```
<?php
$m_shop = '12345'; // merchant ID
$m_orderid = '1'; // invoice number in the merchant's invoicing system
$m_amount = number_format(1, 2, '.', ''); // invoice amount with two decimal places
following a period
$m_curr = 'USD'; // invoice currency
$m_desc = base64_encode('Test'); // invoice description encoded using a base64
algorithm
$m_key = 'Your secret key';

$arHash = array(
    $m_shop,
    $m_orderid,
    $m_amount,
    $m_curr,
    $m_desc
);

// Adding additional parameters if you have entered them
if (isset($m_params))
{
    $arHash[] = $m_params;
}

// Adding the secret key
$arHash[] = $m_key;

// Forming a signature
$sign = strtoupper(hash('sha256', implode(":", $arHash)));
```

Python 2

```
import binascii
from hashlib import sha256

m_shop = "12345"
m_orderid = "1"
m_amount = "1.00"
m_curr = "USD"
description = "Test"
m_desc = binascii.b2a_base64(description.encode('utf8'))[:-1].decode()
m_key = "Secret key"

list_of_value_for_sign = map(str, [m_shop, m_orderid, m_amount, m_curr, m_desc,
m_key])

result_string = ":".join(list_of_value_for_sign)

sign_hash = sha256(result_string)

sign = sign_hash.hexdigest().upper()
```

C#

```
using System;
using System.Security.Cryptography;
using System.Text;

namespace Rextester
```

```
{
    public class Program
    {
        public static void Main(string[] args)
        {
            var m_shop = "12345";
            var m_orderid = "1";
            var m_amount = "1.00";
            var m_curr = "USD";
            var m_desc = Base64Encode("Test");
            var m_key = "Secret key";

            var arr = new string[] { m_shop, m_orderid, m_amount, m_curr, m_desc, m_key };

            var sign = sign_hash(String.Join(":", arr));
        }

        public static string sign_hash(string text)
        {
            byte[] data = Encoding.Default.GetBytes(text);
            var result = new SHA256Managed().ComputeHash(data);
            return BitConverter.ToString(result).Replace("-","").ToUpper();
        }

        public static string Base64Encode(string plainText)
        {
            var plainTextBytes = System.Text.Encoding.UTF8.GetBytes(plainText);
            return System.Convert.ToBase64String(plainTextBytes);
        }
    }
}
```

Payment Processor

The payment processor is designated to process payments on your website. This page is where orders can be marked as paid or, for example, funds can be sent to the customer's account.

Sample Payment Processor

PHP

```
<?php
// Rejecting queries from IP addresses not belonging to Payeer
if (!in_array($_SERVER['REMOTE_ADDR'], array('185.71.65.92', '185.71.65.189',
'149.202.17.210'))) return;

if (isset($_POST['m_operation_id']) && isset($_POST['m_sign']))
```

```
{
        $m_key = 'Your secret key';

        // Forming an array for signature generation
        $arHash = array(
                $_POST['m_operation_id'],
                $_POST['m_operation_ps'],
                $_POST['m_operation_date'],
                $_POST['m_operation_pay_date'],
                $_POST['m_shop'],
                $_POST['m_orderid'],
                $_POST['m_amount'],
                $_POST['m_curr'],
                $_POST['m_desc'],
                $_POST['m_status']
        );

        // Adding additional parameters to the array if such parameters have been
transferred
        if (isset($_POST['m_params']))
        {
                $arHash[] = $_POST['m_params'];
        }

        // Adding the secret key to the array
        $arHash[] = $m_key;

        // Forming a signature
        $sign_hash = strtoupper(hash('sha256', implode(':', $arHash)));

        // If the signatures match and payment status is "Complete"
        if ($_POST['m_sign'] == $sign_hash && $_POST['m_status'] == 'success')
        {
                // Here you can mark the invoice as paid or transfer funds to your customer

                // Returning that the payment was processed successfully
                ob_end_clean(); exit($_POST['m_orderid'].'|success');
        }

        // If not, returning an error
        ob_end_clean(); exit($_POST['m_orderid'].'|error');
}
```

⚠ The payment processor must return m_orderid with a status of “success” or “error” as indicated in the above example. Otherwise a notification will be sent until a correct response has been obtained.

To ensure the security of queries accepted from your website, be sure to install a check of the sender’s IP address. You can get a list of our IP addresses here https://payeer.com/merchant/ips.txt

You can always see how your payment processor has responded to our payment notification by going to History, clicking on your merchant’s tab, and then clicking on the notification button.

Descriptions of Payment Processor Parameters

Name	Key	Description
Internal payment number	***m_operation_id***	A unique payment number in the Payeer system **Example:** 123456
Method of payment	***m_operation_ps***	The ID of the method of payment used to make the payment. All methods of payment can be viewed in the “Appearance” tab under merchant settings
Date and time of transaction	***m_operation_date***	The date and time of the transaction (UTC+3) **Example:** 21.12.2012 21:12
Date and time of payment	***m_operation_pay_date***	The date and time of the payment (UTC+3) **Example:** 21.12.2012 21:12
Merchant ID	***m_shop***	The merchant ID registered in the Payeer system
Payment ID	***m_orderid***	The order ID following the merchant’s invoicing system **Example:** 12345
Payment amount	***m_amount***	The amount of the payment **Example:** 1.00
Payment currency	***m_curr***	The currency used in the payment

		Potential currencies: USD, EUR, RUB
Payment description	***m_desc***	A description of the product or service encoded using a base64 algorithm **Example:** dGVzdA==
Payment status	***m_status***	The status of the payment in the Payeer system **Values used:** success
Electronic signature	***m_sign***	A security signature used to check the cohesiveness of the information obtained and directly identify the sender **Example:** 9F86D081884C7D659A2FEAA0C55AD015A3BF4F1B2B0B822CD15D6C15B0F00A08
Customer’s email address	***client_email***	The email address of the customer who paid the invoice **Example:** customer@email.com
Customer's account number	*client_account*	The Payeer account number of the customer who paid the invoice **Example:** P1000001
Transfer ID	***transfer_id***	The transaction ID of the transfer to the merchant's account for this payment **Example:** 12345
Transfer amount	***summa_out***	The amount of money, minus all fees, transferred to the merchant's account for this payment

		Example: 1.00
Additional parameters	*m_params*	A JSON array of data of additional parameters