

**PAYEER**

## Подключение

Для подключения необходимо:

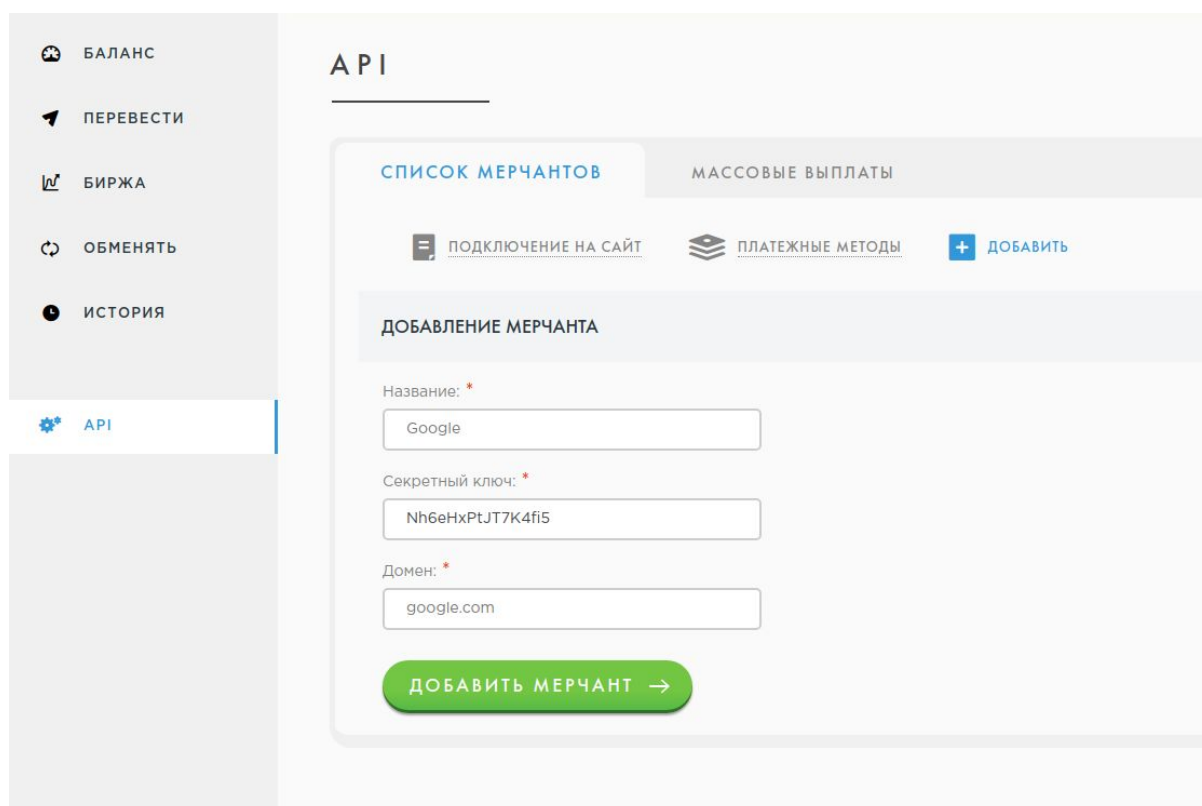
1. [Зарегистрироваться](#) в системе Payeer

2. Активировать Мерчант на вкладке “[Настройка мерчанта](#)”, для этого необходимо перейти на вкладку “Активация” и заполнить следующие поля:

*Название* - имя Вашего сайта, например “Google”, данное название будет отображаться как в Вашем списке мерчантов, так и у пользователя при оплате счета.

*Секретный ключ* - строка символов для подписи данных, передаваемых в форме инициализации платежа и для проверки поступивших данных в обработке платежа.

*Домен* - домен Вашего сайта, например “google.com”, необходимо использовать только латинские символы, а цифры и знак дефиса, для конвертации национальных доменов, необходимо использовать любой punycode-конвертор, например [punycoder](#).



The screenshot shows the 'API' section of a web interface. On the left is a sidebar with navigation items: БАЛАНС, ПЕРЕВЕСТИ, БИРЖА, ОБМЕНЯТЬ, ИСТОРИЯ, and API (highlighted). The main content area is titled 'API' and contains two tabs: 'СПИСОК МЕРЧАНТОВ' (active) and 'МАССОВЫЕ ВЫПЛАТЫ'. Below the tabs are three buttons: 'ПОДКЛЮЧЕНИЕ НА САЙТ', 'ПЛАТЕЖНЫЕ МЕТОДЫ', and '+ ДОБАВИТЬ'. The 'ДОБАВИТЬ' button is highlighted. Below this is a form titled 'ДОБАВЛЕНИЕ МЕРЧАНТА' with three input fields: 'Название:' (containing 'Google'), 'Секретный ключ:' (containing 'Nh6eHxPtJT7K4fi5'), and 'Домен:' (containing 'google.com'). A green button labeled 'ДОБАВИТЬ МЕРЧАНТ →' is at the bottom of the form.

3. Подтвердить владение сайтом через размещение текстового файла в корневой директории Вашего сайта.

#### 4. Заполнить поля:

*Success URL* - на данный адрес клиент будет перенаправлен после успешной оплаты.

*Fail URL* - этот адрес используется для перенаправления в случае ошибки в процессе оплаты или отмены платежа.

*Status URL* - адрес обработчика платежа, на данной странице заказ должен помечаться как оплаченный или, например, происходить зачисление денег на счет клиента на Вашем сайте.

*Ключ для шифрования дополнительных параметров* - секретный ключ для шифрования дополнительных полей, а также динамических success, fail и status url.

5. Отправить мерчант на модерацию. До модерации оплату может производить только владелец мерчанта, при условии, что до оплаты он уже авторизован в Payeer.

## Настройка мерчанта

На вкладке “Настройки” возможно:

1. Задать новый секретный ключ (к сожалению, посмотреть старый секретный ключ невозможно в целях безопасности, но Вы всегда можете изменить его на новый).

2. Установить плательщика комиссии:

2.1. По-умолчанию: клиент платит комиссию выбранного платежного метода, мерчант - комиссию Payeer (по умолчанию: 0.95 %), на счет мерчанта зачисляется сумма счета минус 0.95 %.

2.2. Клиент: комиссию платежного метода и комиссию Payeer платит клиент, на счет мерчанта зачисляется сумма счета полностью.

2.3. Мерчант: комиссию платежного метода и комиссию Payeer платит мерчант, на счет мерчанта зачисляется сумма за вычетом двух комиссий.

3. Задать ключ для шифрования дополнительных параметров, если Вы планируете их передавать в форму оплаты или Вам нужны динамические адреса взаимодействия (успешной, неуспешной оплаты или адрес обработчика).

4. Задать адреса успешной, неуспешной оплаты и адрес обработчика платежа.

5. Выбрать сервер уведомлений (по умолчанию, при отправке уведомления происходит автоматический выбор).

На вкладке “Внешний вид” можно отключить ненужные разделы, методы оплаты или оставить только нужные валюты. Серым цветом выделены методы, недоступные для прямой оплаты счета, но Ваш клиент всегда может пополнить внутренний счет Payeer, а затем оплатить Ваш счет с аккаунта Payeer, для этого после клика на такие методы предоставлена подробная инструкция с возвратом к оплате Вашего счета.

На вкладке “Подключение на сайт” можно скачать готовый модуль для CMS и найти примеры для ручного подключения.

Для тестирования сгенерированной подписи можно использовать одноименную вкладку в настройках мерчанта.

## Форма инициализации платежа

Для перехода на страницу оплаты необходимо сформировать следующую форму:

```
<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>
```

## Расшифровка параметров формы инициализации платежа

Название	Ключ	Описание
URL мерчанта	<i>action</i>	URL-адрес мерчанта
Идентификатор мерчанта	<i>m_shop</i>	Идентификатор мерчанта
Идентификатор платежа	<i>m_orderid</i>	<p>В этом поле продавец задает идентификатор покупки в соответствии со своей системой учета. Желательно использовать уникальный номер для каждого платежа. Идентификатор должен представлять собой любую строку длиной не более 32 символов "A-z", "_", "0-9", "-".</p> <p><b>Пример:</b> 12345</p>
Сумма платежа	<i>m_amount</i>	<p>Сумма платежа, которую продавец желает получить от покупателя. Сумма должна быть больше нуля, дробная часть отделяется точкой, количество знаков после точки - два знака.</p> <p><b>Пример:</b> 1.00</p>
Валюта платежа	<i>m_curr</i>	<p>Валюта платежа</p> <p><b>Возможные валюты:</b> USD, EUR, RUB</p>
Описание платежа	<i>m_desc</i>	<p>Описание товара или услуги. Формируется продавцом. Строка добавляется в назначение платежа.</p> <p>Кодируется алгоритмом base64.</p> <p><b>Пример:</b> dGVzdA==</p> <p><b>PHP код:</b> &lt;?php echo</p>

		base64_encode('test'); ?>
Электронная подпись	<i>m_sign</i>	Контрольная подпись, которая используется для проверки целостности полученной информации и однозначной идентификации отправителя.  <b>Пример:</b> 9F86D081884C7D659A2FE AA0C55AD015A3BF4F1B2 B0B822CD15D6C15B0F00 A08
ID платежной системы	<i>form[ps]</i>	ID платежной системы для автоматического выбора. Список ID можно увидеть на вкладке <b>Внешний вид</b>  <b>Пример:</b> 2609
Валюта платежной системы	<i>form[curr[psld]]</i>	Валюта платежной системы  <b>Пример:</b> USD
Дополнительные параметры	<i>m_params</i>	JSON массив данных дополнительных параметров, зашифрованный с помощью алгоритма Rijndael-256 и закодированный с помощью алгоритма base64
Метод шифрования	<i>m_cipher_method</i>	Метод шифрования m_params. Доступные методы: <ul style="list-style-type: none"> <li>• AES-256-CBC</li> <li>• AES-256-CBC-IV</li> </ul>

#### Дополнительные параметры

Название	Ключ	Описание
Success URL	<i>success_url</i>	на данный адрес клиент

		будет перенаправлен после успешной оплаты
Fail URL	<i>fail_url</i>	этот адрес используется для перенаправления в случае ошибки в процессе оплаты или отмены платежа
Status URL	<i>status_url</i>	адрес обработчика платежа, на данной странице заказ должен помечаться как оплаченный или, например, происходить зачисление денег на счет клиента на Вашем сайте
Дополнительные поля	<i>reference</i>	Массив дополнительных полей, которые необходимо вернуть на обработчик платежа
Домен сабмерчанта	<i>submerchant</i>	Домен сабмерчанта (только для агрегаторов)

Пример создания формы инициализации платежа

#### PHP

```
<?php
$m_shop = '12345'; // id мерчанта
$m_orderid = '1'; // номер счета в системе учета мерчанта
$m_amount = number_format(100, 2, '.', ''); // сумма счета с двумя знаками после точки
$m_curr = 'USD'; // валюта счета
$m_desc = base64_encode('Test'); // описание счета, закодированное с помощью
алгоритма base64
$m_key = 'Ваш секретный ключ';

// Формируем массив для генерации подписи
$arHash = array(
    $m_shop,
    $m_orderid,
    $m_amount,
    $m_curr,
    $m_desc
);

/*
```

```

// Формируем массив дополнительных параметров
$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',

    // Формируем массив дополнительных полей
    'reference' => array(
        'var1' => '1',
        'var2' => '2',
        'var3' => '3',
        'var4' => '4',
        'var5' => '5',
    ),

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

// Формируем ключ для шифрования
$key = md5('Ключ для шифрования дополнительных параметров'.$m_orderid);

// Шифруем дополнительные параметры
//$m_params = urlencode(base64_encode(mcrypt_encrypt(MCRYPT_RIJNDAEL_256,
$key, json_encode($arParams), MCRYPT_MODE_ECB)));

// Шифруем дополнительные параметры с помощью AES-256-CBC (для >= PHP 7)
$iv = substr(hash('sha256', $key), 0, 16);
$m_params = urlencode(base64_encode(openssl_encrypt(json_encode($arParams),
'AES-256-CBC', $key, OPENSSSL_RAW_DATA, $iv)));

// Добавляем параметры в массив для формирования подписи
$arHash[] = $m_params;
*/

// Добавляем в массив для формирования подписи секретный ключ
$arHash[] = $m_key;

// Формируем подпись
$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',
    );

    $url = 'https://payeer.com/merchant?'.http_build_query($arParams);
    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 = "Ваш секретный ключ";

        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();
}
}

```

## Формирование цифровой подписи

Цифровая подпись представляет собой строку символов, которая состоит из значений передаваемых переменных, с добавлением в конец секретного ключа, разделенных символом “.” (двоеточие) и зашифрованных с помощью алгоритма SHA-256. Все буквы преобразуются в прописные.

Пример формирования подписи

### PHP

```

<?php
$m_shop = '12345'; // id мерчанта
$m_orderid = '1'; // номер счета в системе учета мерчанта
$m_amount = number_format(100, 2, '.', ''); // сумма счета с двумя знаками после точки
$m_curr = 'USD'; // валюта счета
$m_desc = base64_encode('Test'); // описание счета, закодированное с помощью

```

```

алгоритма base64
$m_key = 'Ваш секретный ключ';

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

// Добавляем доп. параметры, если Вы их задали
if (isset($m_params))
{
    $arHash[] = $m_params;
}

// Добавляем секретный ключ
$arHash[] = $m_key;

// Формируем подпись
$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 = "Секретный ключ"

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 = "Секретный ключ";

            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);
        }
    }
}

```

## Обработчик платежа

Обработчик платежа предназначен для завершения процесса оплаты на Вашем сайте. На данной странице заказ должен помечаться как оплаченный или, например, происходить зачисление денег на счет клиента.

Пример обработчика платежа

### PHP

```

<?php
// Отключаем запросы с IP-адресов, которые не принадлежат 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 = 'Ваш секретный ключ';

    // Формируем массив для генерации подписи
    $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']
    );

    // Если были переданы дополнительные параметры, то добавляем их в
массив
    if (isset($_POST['m_params']))
    {
        $arHash[] = $_POST['m_params'];
    }

    // Добавляем в массив секретный ключ
    $arHash[] = $m_key;

    // Формируем подпись
    $sign_hash = strtoupper(hash('sha256', implode(':', $arHash)));

    // Если подписи совпадают и статус платежа "Выполнен"
    if ($_POST['m_sign'] == $sign_hash && $_POST['m_status'] == 'success')
    {
        // Здесь можно пометить счет как оплаченный или зачислить денежные
средства Вашему клиенту

        // Возвращаем, что платеж был успешно обработан
        ob_end_clean(); exit($_POST['m_orderid'].'|success');
    }

    // В противном случае возвращаем ошибку
    ob_end_clean(); exit($_POST['m_orderid'].'|error');
}

```



ответ.

Обработчик платежа должен обязательно вернуть `m_orderid` со статусом `success` или `error` как указано в примере выше. В противном случае уведомление будет отправляться, пока не будет получен правильный



Для безопасности принимаемых запросов с нашего сайта обязательно установите проверку по IP-адресу отправителя. Список наших IP-адресов Вы можете получить по ссылке <https://payeer.com/merchant/ips.txt>

Вы всегда можете посмотреть, что отвечает Ваш обработчик платежа на наше уведомление об оплате, если зайдете в [Историю](#), кликните на вкладку Вашего мерчанта и нажмете на значок уведомления.

#### Расшифровка параметров обработчика платежа

Название	Ключ	Описание
Внутренний номер платежа	<code>m_operation_id</code>	Уникальный номер платежа в системе «Payeer». <b>Пример:</b> 123456
Способ оплаты	<code>m_operation_ps</code>	Идентификатор способа оплаты с помощью которого была произведена оплата. Все способы оплаты можно посмотреть на вкладке «Внешний вид» в настройках мерчанта.
Дата и время формирования операции	<code>m_operation_date</code>	Дата и время формирования операции (UTC+3) <b>Пример:</b> 21.12.2012 21:12
Дата и время выполнения платежа	<code>m_operation_pay_date</code>	Дата и время оплаты (UTC+3) <b>Пример:</b> 21.12.2012 21:12
Идентификатор мерчанта	<code>m_shop</code>	Идентификатор мерчанта

		зарегистрированного в системе Payeer
Идентификатор платежа	<i>m_orderid</i>	Идентификатор покупки в соответствии с системой учета мерчанта.  <b>Пример:</b> 12345
Сумма платежа	<i>m_amount</i>	Сумма платежа  <b>Пример:</b> 1.00
Валюта платежа	<i>m_curr</i>	Валюта платежа  <b>Возможные валюты:</b> USD, EUR, RUB
Описание платежа	<i>m_desc</i>	Описание товара или услуги закодированное алгоритмом base64.  <b>Пример:</b> dGVzdA==
Статус платежа	<i>m_status</i>	Статус платежа в системе Payeer  <b>Принимаемые значения:</b> success
Электронная подпись	<i>m_sign</i>	Контрольная подпись, которая используется для проверки целостности полученной информации и однозначной идентификации отправителя  <b>Пример:</b> 9F86D081884C7D659A2FE AA0C55AD015A3BF4F1B2 B0B822CD15D6C15B0F00 A08
E-mail клиента	<i>client_email</i>	E-mail клиента, который оплатил данный счет  <b>Пример:</b> client@email.com
Номер аккаунта клиента	<i>client_account</i>	Номер Payeer аккаунта клиента, который оплатил данный счет

		<b>Пример:</b> P1000001
ID зачисления	<i>transfer_id</i>	ID операции зачисления денежных средств на счет мерчанта по данному платежу  <b>Пример:</b> 12345
Сумма зачисления	<i>summa_out</i>	Сумма за вычетом всех комиссий, которая была зачислена на счет мерчанта по данному платежу  <b>Пример:</b> 1.00
Дополнительные параметры	<i>m_params</i>	JSON массив данных дополнительных параметров