PayPlug Developer Documentation Documentation

Release 1.0

PayPlug Team

June 09, 2015

Contents

Table of contents

CHAPTER 1

Overview

The PayPlug Developer Documentation enables developers to install the PayPlug online payment system to any website or e-commerce store, or to build a payment plugin for an existing CMS (Magento, Wordpress, WooCommerce, Prestashop, etc.). If you develop using PHP, you should use the PayPlug PHP library accessible on http://payplug-php.readthedocs.org/.

Payment process

1. Your website generates a dynamic payment URL by passing the payment parameters (e.g., amount, customer ID, return URL)
2. Redirect your customer to that URL, which points to a secure online payment page (see an example on https://www.payplug.fr/exemple)
3. Once the payment goes through, PayPlug redirects your customer towards the return URL you specified
4. Simultaneously, PayPlug sends an Instant Payment Notification (IPN) to your server, inlcuding all transaction data, to confirm that the payment was processed.

Security & encryption

The PHP library enables secure data transmission via public/private RSA key encryption obtained with the OpenSSL library. In addition, credit card data is encrypted and processed exclusively on PCI DSS servers operated by Payline except ATOS. This protocol guarantees a SSL-level security without requiring to setup an SSL certificate on the merchant's website.

CHAPTER 2

Configuration

PayPlug generates a set of unique parameters and keys for each user account, which needs to be saved in on your server by following these configuration instructions.

Your LIVE MODE account parameters are available through the following setup URL (you will be asked to log in with your PayPlug account). See below for TEST (Sandbox) mode.

```
https://www.payplug.fr/portal/ecommerce/autoconfig
```

Note : your account needs to be activated to use the LIVE mode. When you sign up to PayPlug, you can only use the test mode.

To access your TEST (Sandbox) account parameters, you need to access the following URL (more information in the *TEST (Sandbox) Mode* section):

```
https://www.payplug.fr/portal/test/ecommerce/autoconfig
```

We advise you to access this URL directly from your server and save the parameters in a file or in a database. The JSON object contains the following fields:

Name	Description
status	http response code (200 if the request successful).
amount_min	Minimum amount allowed per transaction (EUR).
amount_max	Maximum amount allowed per transaction (EUR).
currencies	Currencies table (ISO 4217 format) allowed for your account.
url	The base URL to use in constructing the payment with the data and signature fields (<base_url> parameters in the "creating a payment" section).
yourPrivateKey	Your RSA key to be used to sign transaction data. Without this signature, PayPlug will refuse the payment request.
payplugPublicKey	The RSA key that allows you to verify notifications that the payment notifications (IPN) sent to your server was sent by PayPlug and was not modified by a third party.

Remark: you must save `yourPrivateKey` and `payplugPublicKey` in the PEM format (base64 representation of a RSA private key). You can find examples of this format below.

- Private key: https://gist.github.com/splanquart/7827287
- Public key: https://gist.github.com/splanquart/7827462

CHAPTER 3

TEST (Sandbox) Mode

PayPlug has created a sandbox environment for testing transactions without making real payments. You can simply enable or disable the test mode by loading the appropriate parameter in the *configuration* section.

When you sign up to PayPlug, you can only use the TEST mode. Once your account is verified, you will be able to configure your site with the LIVE mode (`https://www.payplug.fr/portal/ecommerce/autoconfig`) to make real transactions.

Important: When you want to switch to LIVE or TEST mode, don't forget to run the *configuration* step again with the appropriate URL.

CHAPTER 4

Creating a payment

In order to accept a payment, you must redirect your customer to a URL of the form:

```
<base_url>?data=<data>&sign=<signature>
```

Example in PHP

```php
<?php
$url = $base_url . "?data=".$data . "&sign=".$signature;
header("Location: $url");
exit();
```

The `<base_url>` **parameter**

This is the `url` field retrieved and saved during the configuration step.

The `<data>` **parameter**

The `data` parameter contains the payment data you wish to send to PayPlug. This data must be encoded in base64 then as a URL.

Example in PHP

```php
<?php
$params = array(
          'amount' => 999,
          'currency' => 'EUR',
          'ipn_url' => 'http://www.example.org/ipn.php',
          'email' => 'john.doe@client.com',
          'first_name' => 'John',
          'last_name' => 'Doe'
          );
$url_params = http_build_query($params);
$data = urlencode(base64_encode($url_params));
```

The fields `amount`, `currency` and `ipn_url` are required. Note that if any of the fields `email`, `first_name` or `last_name` is left blank, the customer will be required to enter **all three fields** on the payment page.

The complete list of accepted fields is available in the *reference* section.

The `<signature>` **parameter**

The `signature` parameter is a SHA1 + RSA signature calculated from the data field value (not encoded in base64, i.e. $url_params in the following example).

This signature must be calculated using OpenSSL with the private key you saved during the setup step. The signature must be encoded in base64 and then as a URL.

Example in PHP

```php
<?php
/* $pvkey = yourPrivateKey saved during the configuration phase */
$privatekey = openssl_pkey_get_private($pvkey);
openssl_sign($url_params, $signature, $privatekey, OPENSSL_ALGO_SHA1);
$signature = urlencode(base64_encode($signature));
```

CHAPTER 5

Instant Payment Notification (IPN)

After every successful payment or refund, PayPlug sends an Instant Payment Notification (IPN) as an HTTP POST request to the URL you provided in the `ipn_url` field.

Create a file at the `ipn_url` address that will be requested after each payment. The IPN must be sent to a publicly accessible URL on your site.

You will find the data in the body of the request which is signed by PayPlug. The signature is located in the PayPlug-Signature field in the request header and is calculated in the same way as the payment request (SHA1 + RSA). You should check the signature before taking into account the IPN data using the public key from the setup page.

The following example sends an email to the administrator each time an IPN is received.

Example in PHP

```php
<?php
$headers = getallheaders();
/* For more security, put the keys in uppercase and retrieve
 * the signature using the key in uppercase
 */
$headers = array_change_key_case($headers, CASE_UPPER);
$signature = base64_decode($headers['PAYPLUG-SIGNATURE']);

/* The data is sent in the body of the POST request in JSON format
 * (MIME type = application / json).
 * Example : {"state": "paid", "customer", "2", "transaction_id": 4121, "custom_data": "29", "order"
 */
$body = file_get_contents('php://input');
$data = json_decode($body);

/* $pbkey = PayPlug public key that you saved from the setup page */
$publicKey = openssl_pkey_get_public($pbkey);
$isSignatureValid = (openssl_verify($body , $signature, $publicKey, OPENSSL_ALGO_SHA1) === 1);

/* Take into account the IPN and send an email with the confirmation*/
if($isSignatureValid){
    $message = "IPN received for ".$data->first_name." ".$data->last_name." for an amount of ".$data
    mail("merchant@example.org","IPN Received",$message);
} else {
    mail("merchant@example.org","IPN Failed","The signature was invalid");
}
```

Note that if you have not received the IPN when your client is directed to the confirmation page `return_url`, we advise you to consider that the order is not confirmed to prevent the user to pay again. You should receive the IPN within a few minutes.

If you make payments in TEST (Sandbox mode), the field `is_test` of the IPN will be `true`.

Finally, we recommend you create an `IPN` object to store all notifications received. This will help you retrieve the information in the future.

The complete list of fields sent in the IPN is available in the *reference* section.

CHAPTER 6

Reference

Payment fields

Fields marked with an * are required.

Name	Type	Description
amount *	Integer	Transaction amount, in cents (such as `4207` for 42,07€). We advise you to verify that the amount is between the minimum and maximum amounts allowed for your account.
currency *	String	Transaction currency. Only `EUR` is allowed at the moment.
ipn_url *	String	URL pointing to the `ipn.php` page, to which PayPlug will send payment and refund notifications. This URL must be accessible from anywhere on the Internet (usually not the case in `localhost` environments).
return_url	String	URL pointing to your payment confirmation page, to which PayPlug will redirect your customer after the payment.
cancel_url	String	URL pointing to your payment cancelation page, to which PayPlug will redirect your customer if he cancels the payment.
email	String	The customer's email address.
first_name	String	The customer's first name.
last_name	String	The customer's last name.
customer	String	The customer ID in your database. (limited to 255 characters)
order	String	The order ID in your database. (limited to 255 characters)
custom_data	String	Additional data that you want to receive in the IPN. (limited to 1024 characters)
origin	String	Information about your website version (e.g., 'My Website 1.2') for monitoring and troubleshooting.

IPN fields

Name	Type	Description
state	String	The new state of the transaction: `paid` or `refunded`.
id_transaction	Integer	The PayPlug transaction ID. We recommend you save it and associate it with this order in your database.
amount	Integer	Transaction amount, in cents (such as `4207` for 42,07€).
email	String	The customer's email address, either provided when creating the payment URL or entered manually on the payment page by the customer.
first_name	String	The customer's first name, either provided when creating the payment URL or entered manually on the payment page by the customer.
last_name	String	The customer's last name, either provided when creating the payment URL or entered manually on the payment page by the customer.
customer	String	Customer ID provided when creating the payment URL.
order	String	Order ID provided when creating the payment URL.
custom_data	String	Custom data provided when creating the payment URL.
origin	String	Information about your website version (e.g., 'My Website 1.2 payplug_php0.9 PHP 5.3'), provided when creating the payment URL, with additional data sent by the library itself.
is_test	Boolean	If value is `true`, the payment was done in TEST (Sandbox) mode.

CHAPTER 7

Frequently asked questions

How to test a payment?

A TEST (Sandbox) mode is available to test your setup. See the *TEST (Sandbox) Mode* section to find out more.

The getallheaders function doesn't exist? What should I do?

The PHP function getallheaders() is an alias of the function apache_request_headers(). It works only if you use Apache on your server (also works with FastCGI since PHP 5.4). For other cases, you can find a code example to the following address: https://gist.github.com/splanquart/7826749.

How to call the autoconfig URL from PHP?

You can use the `curl_exec` function of PHP. You can find an example at the following address: https://gist.github.com/splanquart/7827949.