

Step by Step tutorial for using payamigo

Paywall

You should have received `payamigo_paywall_tutorial.zip`

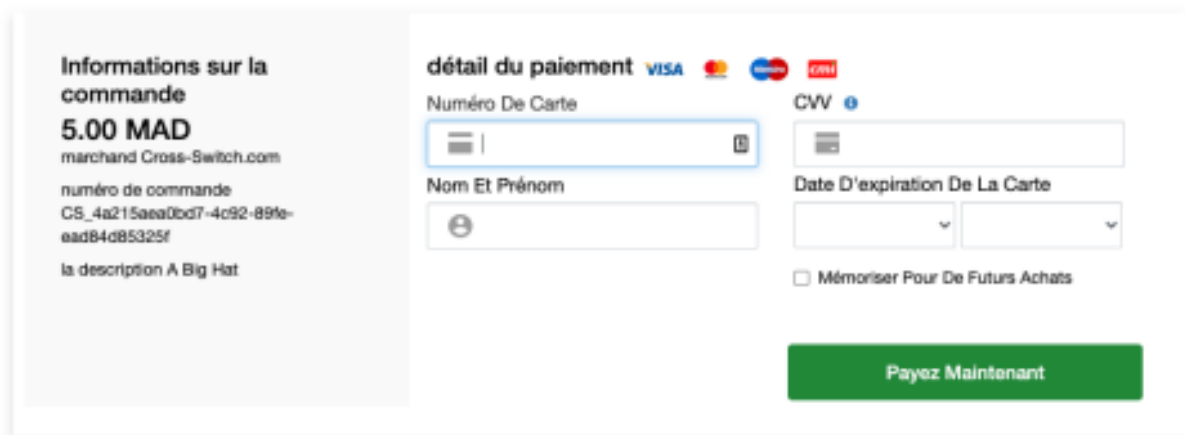
Unpack the Tutorial Example Code

Extract the files and place them in a folder on your web server. For this tutorial, we will be using the following:

The files that do the work:

- **example_index.html** is an iframe that will display the paywall in the customer's browser.
- **launch_paywall.php** encodes and POST's the parameters to the paywall URL to open it.
- **launch_credentials.inc** holds the URL and secret keys required to open the paywall. The provided file is for this demo only. Merchant specific credentials will be provided by our team separately.

Open a browser and browse to **example_index.html** on your server. It should look something like this



The screenshot shows a payment interface. On the left, under 'Informations sur la commande', it displays '5.00 MAD', the merchant 'Cross-Switch.com', a command number, and a description 'A Big Hat'. The main section, 'détail du paiement', includes fields for 'Numéro De Carte', 'CWV', 'Nom Et Prénom', and 'Date D'expiration De La Carte'. It also has a checkbox for 'Mémoriser Pour De Futurs Achats' and a green 'Payez Maintenant' button. At the bottom, there is a security notice in French and logos for Visa, SSL, MasterCard, and others.

Build the Parameter Array

example_index.html has a form that POSTS to **launch_paywall.php** and an iFrame that will display the paywall in the customer's browser.

All of the magic happens in **launch_paywall.php** - it JSON encodes the paywall parameters, signs the payload and POST's the results to the paywall URL to invoke the paywall. This tutorial uses PHP, but you can use any language you want, as long as the end result is an HTTP POST to the paywal URL, with the parameters properly JSON encoded and signed.

The first thing **launch_paywall.php** does is create an array of parameters. A typical case might look like this (replace the hard-coded values with appropriate ones, of course).

```
$payload = array(
    // Authentication parameters
    'merchantAccount' => $merchantAccount,
    'timestamp' => time(),
    'skin' => 'vps-1-vue',

    // Customer parameters
    'customerId' => 'payplus-paywall-poc',
    'customerCountry' => 'MA',
    'customerLocale' => 'en_US',

    // Charge parameters
    'chargeId' => time(), // Optional, unless required by the merchant account
    'price' => '5',
    'currency' => 'MAD',
    'description' => 'A Big Hat',

    // Deep linking
    'mode' => 'DEEP_LINK',
    'paymentMethod' => 'CREDIT_CARD',
    'showPaymentProfiles' => 'true',
    'flowCompletionUrl' => (string)"http://sample.com/hosted-paywall files/redirect.html"
);
```

Refer to the reference section “Input Parameters” for additional launch options.

Encoding the Parameters

The next step is to sign and encode the parameters. The payload is signed using SHA256 and the paywall secret key.

```
$json_payload = json_encode($payload);
$signature = hash('sha256', $paywallSecretKey . $json_payload);
```

POST the encoded payload and signature to the paywall URL

Finally, you POST the encoded parameters to your paywall URL to invoke the paywall:

```
<form id="openPaywall" action="<?php echo $paywallUrl; ?>" method="POST"
target="paywallFrame">
  <input type="hidden" name="payload" value='<?php echo $json_payload; ?>' />
  <input type="hidden" name="signature" value="<?php echo $signature; ?>" />
</form>
```

Note that `$merchantAccount`, `$paywallSecretKey` and `$paywallUrl` are assigned by payamigo.

For the tutorial, we have copied these values into

launch_credentials.inc.

Handling postMessage Events

The paywall sends JavaScript `window.postMessage` event containing a stringified JSON object to communicate back to your web page. This technique is used to avoid cross site scripting issues.

Messages are posted to `window.postMessage`. You can set up a Javascript listener to receive these messages.

To receive and process the message, you can add code like this in your parent page:

```
window.addEventListener("message", receiveMessage, false);

function receiveMessage(event) {
  if (!event || event.origin !== "https://sandbox.cross-switch.dev.com/pwthree/") {
    console.log('Invalid event');
    return;
  }
  // parse event
  var eventObject = JSON.parse(event);
  // Add code here to handle the event
}
```

Refer to the reference section “postMessages Events” for details of the content of each paywall message

Handling Real-Time Notifications

In addition to the postMessage events which you can use to manage your display to the customer, payamigo sends a real-time notification to your server any time a charge changes state. Please provide notification Endpoint on your side for configuration. You should use these notifications to fulfil goods to your customers, and also to be notified of refunds, chargebacks or asynchronous completion events.

Reference

Input Parameters

The payroll supports a number of parameters in addition to those used in this tutorial. To send additional parameters, simply include them in the \$payload array prior to JSON encoding.

Parameter	Required	Description
Authorization Parameters		
merchantAccount	Yes	Your merchant Account Name, provided by payamigo.
timestamp	Yes	The current time measured in the number of seconds since the Unix Epoch (January 1 1970 00:00:00 GMT). This is used to calculate the signature, and is only valid for 30 minutes, limiting exposure to replay attacks.
Charge Parameters		
chargeld	No	A unique value indicating what was purchased, usually your internal transaction ID. It will be passed back to you in the notifications so that you can fulfill the purchase. If you do not provide a chargeld, payamigo will create one for you.
price	Yes	The price is the amount to charge the customer
currency	Yes	the ISO 4217 currency code for the transaction currency (MAD,USD, EUR, etc)

description	Yes	A description of what the customer is buying. This will appear on the UI, on the purchase receipt and in transaction reports.
memo	No	A string that will be recorded with the charge, and included in the memo column of the transaction reports
chargeProperties	No	Optionally adds properties to the charge. Properties are name/value pairs in this format: {"name1":"value1","name2":"value2"}". '=' and ' ' are not allowed in the name. The total size of the string cannot exceed 3500 characters.
Customer Parameters		
customerId	Yes	your ID for this customer. This is used for fraud checks and reporting, and is returned to you so that you can fulfil the purchase.
customerCountry	No	A two-letter ISO country code. If it is required but not provided, the paywall will use IP Geo-location or ask the customer to select their country.
customerName	No	The full name of the customer.
customerEmail	No	The Email address where the customer will receive confirmation emails, if payamigo is sending confirmation emails. If it is required and not provided, the paywall will ask the customer to provide it.

customerStateProvince	No	The state or province of the customer who will be charged. It is needed to calculate tax in some countries. If it is needed and not provided, the paywall will ask the customer to provide it.
customerCity	No	The name of the city the customer lives in.
customerPostalCode	No	The postal code of the customer who will be charged. It is needed for some tax calculations in the US. If it is needed and not provided, the paywall will ask the customer to provide it.
customerStreetAddress	No	The street address of the customer, which may be required to calculate tax on physical goods. If it is required and has not been supplied, the paywall will ask the customer to provide it.
customerStreetAddress2	No	The street address of the customer, which may be required to calculate tax on physical goods. If it is required and has not been supplied, the paywall will ask the customer to provide it.
customerDOB	No	The date of birth for the user, in the format YYYY-MM-DD. If provided it is used for fraud calculations and can be useful when resolving customer complaints. Some payment providers may require a DOB if your product is intended for children.

customerLocale	No	<p>Used to override the default language of the user's country. This is a 2-letter ISO language code and 2-letter ISO country code separated by underscore</p> <p>Supported locales: en_US, fr_FR, pt_BR, ar_EG, da_DK, de_DE, es_ES, fi_FI, it_IT, ja_JP, ko_KR, nl_NL, no_NO, pl_PL, ru_RU, sv_SE, tr_TR</p>
----------------	----	--

<i>Paywall Options</i>		
successUrl	No	The URL the paywall will redirect to after a successful purchase.
failureUrl	No	The URL the paywall will redirect to after a failed purchase attempt, either due to a fraud rejection, inability to communicate with the PSP or a rejection from the PSP.
cancelUrl	No	The URL the paywall will redirect to if the customer cancels the purchase flow.
flowCompletionUrl	No	The URL the paywall will redirect to at the end of the purchase flow for any result if the successUrl, failureUrl or cancelUrl was not specified.
cssUrl	No	An HTTPS URL to custom CSS that will be applied to the paywall when it is opened in a laptop or desktop browser

mobileCssUrl	No	An HTTPS URL to custom CSS that will be applied to the paywall when it is opened on a mobile device
savePaymentProfile	No	Boolean flag, when set to 'true' payment information will automatically be save for offsite methods where we cannot show the opt-out checkbox.
showPaymentProfiles	No	Boolean flag, when set to 'true' Paywall will display previously created stored payment profiles along with new credit cards option.

<i>'Deep Link' mode parameters</i>		
mode	Yes	Set to 'DEEP_LINK' to use deep linking.
paymentMethod	Yes	Should be set to "CREDIT_CARD".

postMessage Events

These are the events the paywall sends via JavaScript `window.postMessage` to communicate back to your web page.

Event	Example	Web page action
Paywall Opened	<pre>{ "name": "PURCHASE_STATU S", "initialized": true }</pre>	If the web page had previously explicitly hidden the paywall IFRAME then it should be unhidden now
Order Submitted	<pre>{ "name": "PURCHASE_STATUS" , "submitted": true }</pre>	This message indicates the charge ID has been consumed but the purchase flow is not yet complete. If the web page is managing its own navigation buttons it should now display 'cancel' instead of 'back'.
Successful Purchase	<pre>{ "name": "PURCHASE_STATUS" , "success": true, "pending": false }</pre>	This message indicates the charge ID has been consumed and the purchase flow has completed. If the web page is managing its own navigation buttons it should now display 'close' instead of 'cancel' or 'back'. If pending is true it indicates the flow is complete but the funds have not yet been captured.

Failed Purchase	<pre>{ "name": "PURCHASE_STATUS", "success": false }</pre>	<p>This message indicates the charge ID has been consumed and the purchase flow has completed. If the web page is managing its own navigation buttons it should now display 'close' instead of 'cancel' or 'back'.</p>
Paywall Closed	<pre>{ "name": "PURCHASE_STATUS", "close": true }</pre>	<p>The web page should hide or remove the paywall IFRAME</p>
Window Resize	<pre>{ "message": { "name": "WINDOW_SIZE", "success": true, "payload": { "size": { "width": 666, "height": 1542 } } } }</pre>	<p>The web page should resize paywall IFRAME to the width and height supplied in the payload.</p>