



Hosted Form Modal

Simple Setup Developer Guide – v1


Contents



1	Hosted Form Modal	3
1.1	Pre-Requisites	4
2	Sample Code	5
2.1	Hosted Form Modal Sample Code (No Lightbox)	5
2.2	Hosted Form Modal Lightbox.....	6



1 Hosted Form Modal



Example Merchant


Purchase Amount: **GBP 10.01**
Order Reference: **Test purchase**

 Test Card



 *****0821 

 02 / 20  123

Verified by  

 **Pay By Card**

Or Pay By

 [Learn More?](#)  [What is PayPal?](#)

1.1 Pre-Requisites

The following is a basic setup for the Hosted Form Modal. It provides the pre-requisites and sample code. Hosted Form Modal is compatible with the features of the standard Hosted Form integration. The author assumes the reader is familiar with the Hosted Form Integration Guide.

The following information is required:

Gateway Merchant ID	<p>Your Merchant ID enables you to access and communicate with the Payment Gateway. Please note that these details will differ to the login supplied to access the administration panel. You should have received these details when your account was set up.</p> <p>You may also use test Merchant IDs (if you have been issued with a test ID) and swap these for your live account details when you receive them.</p>
Integration URL	https://pay.paymentsconnect.co.uk/button/?fields={BASE 64 ENCODED STRING}
If Using Lightbox:	
JQuery	Hosted Form Modal with Lightbox requires JQuery. The Gateway supports 1.8 to 3.4. The code example uses 1.12.4.
Hosted Form JavaScript	https://pay.paymentsconnect.co.uk/sdk/web/v1/js/hostedforms.min.js

2 Sample Code

2.1 Hosted Form Modal Sample Code (No Lightbox)

The following sample code is almost identical to the sample code located in our Hosted Integration Guide. The only difference is the URL has been changed to use the modal endpoint.

```
<?PHP
// Signature key entered on MMS. The demo accounts is fixed to this value,
$key = 'Agency12Also14Entity';
// Gateway URL
$url = 'https://yourURL/hosted/modal/';
if (!isset($_POST['responseCode'])) {
    // Send request to gateway
    // Request
    $req = array(
        'merchantID' => '103237',
        'action' => 'SALE',
        'type' => 1,
        'countryCode' => 826,
        'currencyCode' => 826,
        'amount' => 1001,
        'orderRef' => 'Test purchase',
        'transactionUnique' => uniqid(),
        'redirectURL' => ($_SERVER['HTTPS'] == 'on' ? 'https' : 'http') . '://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI']
    );

    // Create the signature using the function called below.
    $req['signature'] = createSignature($req, $key);
    echo '<form action="' . htmlentities($url) . '" method="post">' . PHP_EOL;
    foreach ($req as $field => $value) {
        echo '<input type="hidden" name="' . $field . '" value="' . htmlentities($value) . '">' . PHP_EOL;
    }
    echo '<input type="submit" value="Pay Now">' . PHP_EOL;
    echo '</form>' . PHP_EOL;
} else {
    // Handle the response posted back
    $res = $_POST;
    // Extract the return signature as this isn't hashed
    $signature = null;
    if (isset($res['signature'])) {
        $signature = $res['signature'];
        unset($res['signature']);
    }

    // Check the return signature
    if (!$signature || $signature !== createSignature($res, $key)) {
        // You should exit gracefully
        die('Sorry, the signature check failed');
    }

    // Check the response code
    if ($res['responseCode'] === "0") {
        echo "<p>Thank you for your payment.</p>";
    } else {
        echo "<p>Failed to take payment: " . htmlentities($res['responseMessage']) . "</p>";
    }
}

// Function to create a message signature
function createSignature(array $data, $key) {
    // Sort by field name
    ksort($data);
    // Create the URL encoded signature string
    $ret = http_build_query($data, '', '&');
    // Normalise all line endings (CRNL|NL|CR) to just NL (%0A)
    $ret = str_replace(array('%0D%0A', '%0A%0D', '%0D'), '%0A', $ret);
    // Hash the signature string and the key together
    return hash('SHA512', $ret . $key);
}
?>
```

2.2 Hosted Form Modal Lightbox

The below example shows 3 different methods of how to use the Lightbox JavaScript. The first button shows how to setup and submit the form manually, whereas the second two buttons show how to submit them automatically using the Hosted Form JavaScript via a JSON payload.

```
<?PHP

// Signature key entered on MMS. The demo accounts is fixed to this value,
$key = 'Agency12Also14Entity';

// Gateway URL
$url = 'https://yourURL/hosted/modal/';

if (!isset($_POST['responseCode'])) {
    // Send request to gateway

    // Request
    $req = array(
        'merchantID' => '103237',
        'action' => 'SALE',
        'type' => 1,
        'countryCode' => 826,
        'currencyCode' => 826,
        'customerName' => 'Test Card',
        'cardNumber' => '4929 4212 3460 0821',
        'cardExpiryMonth' => '02',
        'cardExpiryYear' => '20',
        'cardCVV' => '123',
        'amount' => 1001,
        'duplicateDelay' => 0,
        'orderRef' => 'Test purchase',
        'transactionUnique' => uniqid(),
        'redirectURL' => ($_SERVER['HTTPS'] == 'on' ? 'https' : 'http') . '://' . $_SERVER['HTTP_HOST'] .
        $_SERVER['REQUEST_URI'],
    );

    // Create the signature using the function called below.
    $req['signature'] = createSignature($req, $key);

    // Include JQuery javascript
    echo '<script src="https://code.jquery.com/jquery-1.12.4.js" integrity="sha256-
    Qw82+bXyGq6MydymqBxNPYTaUXXq7c8v3CwiYwLLNXU=" crossorigin="anonymous"></script>' . PHP_EOL;

    // Include HostedForm javascript
    echo '<script src="https://YourURL/sdk/web/v1/js/hostedforms.min.js"></script>' . PHP_EOL;

    // Create form (manually)
    echo '<form id="form1" action="" . htmlentities($url) . "" method="post" target="_blank">' . PHP_EOL;
    foreach ($req as $field => $value) {
        echo '<input type="hidden" name="" . $field . "" value="" .
        htmlentities($value) . "">' . PHP_EOL;
    }
    echo '<input type="submit" value="Pay Now [Manual]">' . PHP_EOL;
    echo '</form>' . PHP_EOL;
    echo '<script>$("#form1").hostedForm();</script>' . PHP_EOL;
}
```

```

// Create form (automatically)
$form2 = array(
    'id' => 'form2',
    'url' => $url,
    'data' => $req,
    'submit' => array(
        'type' => 'text',
        'label' => 'Pay Now [Automatic]',
    ),
);
echo '<div id="form2-holder"></div>';
echo '<script>$("#form2-holder").hostedForm(' . json_encode($form2) . ');</script>' . PHP_EOL;

// Create form (automatically)
$form3 = array(
    'id' => 'form3',
    'url' => $url,
    'data' => $req,
    'attrs' => array(
        'class' => 'form3',
        'title' => 'Form #3',
    ),
    'submit' => array(
        'type' => 'button',
        'label' => '<i>Pay Now [Automatic 2]</i>',
        'attrs' => array(
            'class' => 'btn3',
            'title' => 'Submit Button',
        ),
    ),
);
echo '<div id="form3-holder"></div>';
echo '<script>$("#form3-holder").hostedForm(' . json_encode($form3) . ');</script>' . PHP_EOL;

} else {
    // Handle the response posted back

    $res = $_POST;

    // Extract the return signature as this isn't hashed
    $signature = null;
    if (isset($res['signature'])) {
        $signature = $res['signature'];
        unset($res['signature']);
    }

    // Check the return signature
    if (!$signature || $signature !== createSignature($res, $key)) {
        echo '<p>Sorry, the signature check failed.</p>' . PHP_EOL;

    // Check the response code
    } else if ($res['responseCode'] === '0') {
        echo '<p>Thank you for your payment.</p>' . PHP_EOL;

    // Else failed
    } else {
        echo '<p>Failed to take payment: ' . htmlentities($res['responseMessage']) . '</p>' . PHP_EOL;
    }

    echo '<p><button onclick="window.history.back();">Back</button></p>';
}

echo '</body></html>';

```

```
// Function to create a message signature
function createSignature(array $data, $key) {
    // Sort by field name
    ksort($data);

    // Create the URL encoded signature string
    $ret = http_build_query($data, '', '&');

    // Normalise all line endings (CRNL|NL|CR) to just NL (%0A)
    $ret = str_replace(array('%0D%0A', '%0A%0D', '%0D'), '%0A', $ret);

    // Hash the signature string and the key together
    return hash('SHA512', $ret . $key);
}
```

```
?>
```