



# SDK Developer Reference Manual

Version 1.0

# Table of contents

<b>Overview .....</b>	<b>4</b>
What is the HiPay Mobile SDK.....	4
<i>What is an SDK?</i> .....	4
<i>What is a REST Web Service?</i> .....	4
Who may use this SDK? .....	4
<i>Knowledge and skills</i> .....	4
How to read this documentation? .....	5
<b>Starting Guide.....</b>	<b>6</b>
Basic Concept.....	6
<i>Requests</i> .....	6
<i>Responses</i> .....	6
<i>Character encoding</i> .....	8
<i>Dates</i> .....	8
Available formats .....	9
<i>How to specify the return format?</i> .....	9
XML.....	9
JSON .....	10
Merchant Authentication .....	11
Overview.....	11
Get an API key.....	11
<i>Signatures</i> .....	12
<b>After payment is received (Forward URL) .....</b>	<b>13</b>
Overview.....	13
<i>Response</i> .....	13
<b>Notification of payment (notification URL) .....</b>	<b>14</b>
Overview.....	14
<i>Response</i> .....	14
<i>Status codes</i> .....	16
<b>Notification Response (notification URL).....</b>	<b>17</b>
Overview.....	17

<i>Parameters</i> .....	17
<b>Signature verification</b> .....	<b>18</b>
Check the signature of a response from the API .....	18
Check the signature of a notification .....	20
<i>Signature verification example</i> .....	20
<b>Testing</b> .....	<b>21</b>
Virtual bank card .....	21
Test codes .....	21
Free codes.....	22
More questions? .....	22

# Overview

## What is the HiPay Mobile SDK

<b>Introduction</b>	HiPay Mobile provides access to key features of its payment engine through an API or Web Service.
<b>Web Services types</b>	There are several types of Web Services, like <ul style="list-style-type: none"><li>• REST,</li><li>• XML-RPC,</li><li>• SOAP,</li><li>• etc.</li></ul> HiPay Mobile offers an API based on the most common type, <i>REST</i> .
<b>Advantages</b>	For quicker integration of the API, HiPay Mobile provides to the merchants an SDK that let the merchant use all the API methods with less coding.

## What is an SDK?

<b>Description</b>	A software development kit (SDK or "devkit") is typically a set of software development tools.
<b>Use</b>	This SDK allows for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar platform.

## What is a REST Web Service?

<b>Description</b>	REST (Representational State Transfer) is a way to build Web services. This is neither a protocol nor a format but a style of architecture. Systems that follow the REST principles are based simply on the architectural style of the Web, mainly revolving around the HTTP protocol.
--------------------	--

## Who may use this SDK?

This SDK is opened to all merchants who wish to dynamically integrate the HiPay Mobile payment solution in their sites or applications.

This document is specifically aimed to developers.

## Knowledge and skills

The HiPay Mobile SDK requires that the user is familiar with the following:

- Basic understanding of Web Services (<http://www.w3schools.com/webservices>),
- Manipulating an XML document (<http://www.w3schools.com/xml>)
- Mastery of a programming language allowing the use of Web Services.

## How to read this documentation?

---

### Description

This documentation is devoted to explaining the basic concepts of the SDK:

- How to create a request,
- how to interpret a response,
- the different formats available,
- Authentication and data types used.

### Reference

---

To get in detail the capabilities of each SDK API methods please refer to the “API Reference” document.

---

# Starting Guide

## Basic Concept

---

**Description** The HiPay Mobile SDK consists of an application programming interface (API) in the form of some files to interface to a particular programming language.

In the HiPay Mobile SDK, applications communicate with the server through URLs included in each function.

---

### Requests

---

**Description** All requests to the SDK are sent to the correct API REST URL already defined in each SDK function.

---

**SDK accessibility** The SDK API is accessible via both HTTP and HTTPS.

However, for security reasons, we strongly recommend using the HTTPS protocol in production environments.

---

### Example

For example, to retrieve information about the product identified by 123456 the request would be:

---

### PHP5 SDK

```
require_once 'alopass-apikit-php5/api/AllopassAPI.php';
$api = new AllopassAPI();
$response = $api->getProduct('123456');
var_dump($response);
```

---

### Java SDK

```
AllopassAPI api = new AllopassAPI();
ProductDetailResponse response =
(ProductDetailResponse)api.getProduct(123456);
System.out.println(response.getName());
```

---

### C# SDK

```
AllopassAPI api = new AllopassAPI();
ProductDetailResponse response =
(ProductDetailResponse)api.getProduct(123456);
Console.WriteLine(response.getName());
```

---

## Responses

---

**Description** The execution of an action on the server brings the return of formatted output, with an HTTP status code.

---

### HTTPS

**description** The HiPay Mobile API can potentially return the following HTTP status codes to the SDK:

<b>HTTP Status</b>	<b>Description</b>
<b>200 OK</b>	Everything went well.
<b>201 CREATED</b>	A new resource has been created correctly. For example, a transaction, product, etc.
<b>304 NOT MODIFIED</b>	The content of the response has been cached and has not changed since the last request.
<b>400 BAD REQUEST</b>	One of the parameters of the request is invalid.
<b>401 UNAUTHORIZED</b>	Authentication failed. The API key or signature is invalid.
<b>403 FORBIDDEN</b>	Access to resources is prohibited. For example, this status will be returned when a merchant attempts to retrieve information about a product that they don't own.
<b>404 NOT FOUND</b>	Resource not found.
<b>405 METHOD NOT ALLOWED</b>	The HTTP method (GET, POST, PUT, DELETE) is invalid for the requested resource. For example, requesting the deletion of an account is always a bad idea.
<b>500 SERVER ERROR</b>	An unexpected error occurred. Please report the incident.
<b>503 SERVICE UNAVAILABLE</b>	Access to a resource is temporarily unavailable. The user may have to try again later.

**Response format**

---

Responses are formatted in XML or JSON.  
Each response consists of

- a numeric status code specific to the API,
- a message describing the status and
- a content described in the selected format (whenever information needs to be retrieved).

---

**Request action**

Performed correctly:  
➔ the API will always return code "0" (zero) with the message "OK".

Performed with error:  
➔ the API returns a status code greater than zero with a message describing the error

(For a list of error codes, see Appendix 1 "Error Codes" in the "API Reference document").

---

### Example

For example, to retrieve information about the product identified by 123456 the request would be:

The following is a XML return example.

---

#### XMLreturn

```
<response code="O" message="OK">
[...] CONTENT OF RESPONSE [...]
</response>
```

---

### Character encoding

---

#### Important

All data must be encoded in UTF-8 (Unicode).



#### Verification

Verification of valid UTF-8 is performed. If an invalid sequence is found, it is automatically converted to UTF-8.

---

#### Dates

##### Description

All dates are converted to GMT and the merchants are responsible for the formatting in their preferred time zone.

---

##### Formats

Dates are returned by the API in two different formats as follows:

Format	Description
Timestamp	This is a UNIX timestamp, unsigned integer representing the number of seconds since January 1, 1970 GMT.
ISO-8601	The international standard ISO 8601 specifies numeric representations of date and time.

### Example

The following is a XML return example.

---

#### XML API return

```
<date timestamp="1258387836" date="2009-11-16T16:10:36+00:00" />
```

---

#### Example in PHP

To get this date in PHP

```
<?php
date_default_timezone_set('UTC');

// Display the current date (ISO 8601)
print date(DATE_ISO8601);

// Display the date returned by the API (ISO 8601)
$timestamp=1258387836;
print date(DATE_ISO8601, $timestamp);
```

---

## Available formats

The HiPay Mobile API provides two formats: XML and JSON, each being provided for specific cases of programming.

### How to specify the return format?

#### Examples

The following are examples of requests with the *format* parameter explicitly stated:

---

#### Response

```
$response = $api->getProduct('123456', array('format' =>
'json'), false);
$response = $api->getWebsite(array('id' => '123456',
'format' => 'json'), false);
```

---

## XML

All responses are encapsulated in the <response> XML markup followed by two attributes *code* and *message* containing the code and status message returned by the API, as described under "Basic Concepts>Responses".

#### Example

---

#### XML Response

```
<?xml version="1.0" encoding="UTF-8" ?>
```

---

```
<response xmlns="https://api.allopass.com/rest" code="0"
message="OK">
<id>123456</id>
<name><![CDATA[PRODUCT NAME]]></name>
<purchase_url><![CDATA[http://localhost/purchase]]></purcha
se_url>
<forward_url><![CDATA[http://localhost/product]]></forward_
url>
</response>
```

---

## JSON

JSON is a data format that reuses elements of JavaScript syntax. This greatly facilitates the use of content in an HTML page. This format is typically used to manipulate data in AJAX.

### *Example*

#### **Description**

---

The JSON responses returned by the API are simple conversions of XML strings to JSON. This format is made available in order to provide greater comfort for AJAX application developers.

---

#### **JSON Response**

```
<?xml version="1.0" encoding="UTF-8" ?>
<response xmlns="https://api.allopass.com/rest" code="0"
message="OK">
<id>123456</id>
<name><![CDATA[PRODUCT NAME]]></name>
<purchase_url><![CDATA[http://localhost/purchase]]></purcha
se_url>
<forward_url><![CDATA[http://localhost/product]]></forward_
url>
</response>
```

---

# Merchant Authentication

## Overview

The HiPay Mobile API methods called by the SDK require merchant authentication. The authentication mechanism is based on a set of keys and hash functions (MD5 or SHA-1) to "sign" all exchanges with HiPay Mobile and ensure their authenticity.

## Get an API key

### Description

Merchants must have an API key to use the HiPay Mobile API methods.



This "public" key is unique and identifies each merchant.

In addition to the API key, merchants also have a secret key that allows them to calculate the signature of each request and verify the signature of each response returned by the API.

### Keyset security

This keyset is available in your HiPay Mobile back office under the « Toolbox » category.

Merchants may always generate a new secret key clicking on the "Generate a secret key" button.

This key is randomly generated and published immediately.

To secure communication with the API, it is advisable to regularly change the secret key.

### Warning



A merchant already using the HiPay Mobile API should be particularly careful in generating a new secret key. New keys are automatically taken into account by the API and any request is refused until the key has been updated on the applications and merchant sites.

## Set your account in the HiPay Mobile SDK

---

**Description** To let your account use the HiPay Mobile SDK, you must edit the “\conf\conf.xml” file and fill the “account” parameters this way :

---

**JSON Response**

```
<accounts>
  <account email="youremail@mail.com">
    <keys>
      <api_key>01bc7a393c0cb427c0c4d96e5c09890</api_key>

      <private_key>4e06989e52c2bd5b569f4c55ac76546</private_key>
    >
    </keys>
  </account>
</accounts>
```

---

## Signatures

---

**Authentication steps** Authentication happens at three steps while using the HiPay Mobile API

Steps	Description
<b>1-Building Requests</b>	This is a UNIX timestamp, unsigned integer representing the number of seconds since January 1, 1970 GMT.
<b>2-Verification of responses</b>	The response to a signed request is also signed. The API returns a signature calculated from merchant identifiers (API keys). Verifying the validity of the signature confirms that the response comes from HiPay Mobile. This verification is optional and is at the merchant's discretion. (see “Signature Verification”)
<b>3-Verification of notifications</b>	All events and payment notifications sent to a merchant are signed using the merchant's API keys. It is strongly advised to verify the signature and validate that HiPay Mobile is the source of the notification. (see “Signature Verification”)

---

# After payment is received (Forward URL)

## Overview

---

**Description** After a successful payment the Forward URL passes GET parameters that you can collect and use to query the */transaction* for verification.

---

## Response

The following table lists the parameters returned to the merchant URL.

Parameter	Description	Example
<b>data</b>	custom data that was initially passed to the <i>transaction/prepare</i> API.	Payment for 5 widgets
<b>code[]</b>	HiPay Mobile access code. If several codes were required for purchase, the list of codes is comma-delimited.	KFD45
<b>transaction_id</b>	HiPay Mobile ID of the transaction	0c92578d-3143-4bd8-aeae-72f2455e2499
<b>merchant_transaction_id</b>	optional merchant transaction ID initially passed to the <i>transaction/prepare</i> API.	ABC123DEF456
<b>DATAS</b>	<i>same as data [deprecated and only intended for backward compatibility]</i>	
<b>RECALL</b>	<i>same as code [deprecated and only intended for backward compatibility]</i>	
<b>codes[]</b>	<i>same as code [deprecated and only intended for backward compatibility]</i>	
<b>trxid</b>	<i>same as transaction_id [deprecated and only intended for backward compatibility]</i>	

# Notification of payment (notification URL)

## Overview

### Description

---

When the *url\_notification* parameter is set in the product or in the transaction (*transaction/prepare*), then merchants are registered to receive acknowledgments of their transactions on a URL placed on their server.

---

### Response

The following table lists the parameters returned to the merchant URL.

Parameter	Description	Example
<b>action</b>	Describes the type of event for which we notify (always <i>payment-confirm</i> for payments received)	payment-confirm
<b>test</b>	Allows merchant to identify test transactions.	<i>true or false</i>
<b>transaction_id</b>	Unique ID for the HiPay Mobile transaction.	0c92578d-3143-4bd8-aeae-72f2455e2499
<b>status</b>	Status of the transaction.	0
<b>status_description</b>	Description of the status of the transaction.	Success
<b>access_type</b>	Product type	onetime-dynamic
<b>date</b>	Transaction date	2010-12-15T16:09:57+00:00
<b>code</b>	HiPay Mobile code used	XXXXXXXXXX
<b>pricepoint_id</b>	Pricepoint Id used	206
<b>type</b>	Payment method	premium-sms
<b>data</b>	Custom merchant data provided when making payment request.	Widget592
<b>merchant_transaction_id</b>	The transaction ID used by the merchant.	IN1237123
<b>amount</b>	Amount of the transaction. The <i>currency</i> is determined by the price	10.00

Parameter	Description	Example
	point identifier.	
<b>paid</b>	Amount actually paid by the customer in <i>currency</i> .	10.00
<b>currency</b>	Local currency for the transaction.	EUR
<b>payout_amount</b>	Amount of the merchant payout in payout currency	6.18
<b>payout_currency</b>	Payout currency for the transaction	EUR
<b>reference_currency</b>	Base currency (EUR is default).	USD
<b>reference_amount</b>	Amount of the transaction in base currency	14.79
<b>reference_paid</b>	Amount actually paid by the customer in base currency	14.79
<b>reference_payout</b>	Amount of the merchant payout in base currency	9.14
<b>customer_country</b>	Country code of the customer. This two-letter code complies with ISO-3166.	FR
<b>site_id</b>	Identifier of the merchant site.	123456
<b>product_name</b>	The name of the product where the code was used	My New Product
<b>api_key</b>	Merchant API key	cfd3b9a6b7b309c06aa53f55 27c96e67
<b>api_ts</b>	UNIX Timestamp (GMT)	1258691527
<b>api_hash</b>	Hash function to use for signing: <i>md5</i> or <i>sha1</i> . (SHA-1 is default)	sha1 md5
<b>api_sig</b>	Signature	fb1bab50fd2c3751dab07b35 ...

### Example

#### URL notification Example

---

```
http://your-domain.com/allopass_notification.php?action=payment-confirm
&transaction_id=0c92578d-3143-4bd8-aeae-
72f2455e2499&status=0&status_description=success&data=
&merchant_transaction_id=&amount=10.00&paid=10.00&currency=EUR&reference_curren
cy=USD&reference_amount=14.79
&reference_paid=14.79&reference_payout=9.14&payout_currency=EUR&payout_amount=6.
18&customer_country=FR
&site_id=123456&api_hash=sha1&api_ts=1258691527&api_key=cfd3b9a6b7b309c06aa53f55
27c96e67&api_sig=1c90d5846d16f7f9fede3ff3d6769193fe5b0d1a
```

---

### Status codes

Code	Description	Example
0	Success	Payment accepted

# Notification Response (notification URL)

## Overview

---

<b>Description</b>	Merchants may return an XML response to HiPay Mobile so as to explicitly acknowledge that the notification was successfully processed.
<b>Notification response</b>	<p>The Notification Response is optional but recommended. In the absence of a response from the merchant, HiPay Mobile will interpret an HTTP 200 return code as a success.</p> <p>A response status of 0 is a failure, 1 is a success. After a failure, HiPay Mobile will attempt 4 more times.</p>

---

## Parameters

The following table lists the parameters utilized in code. They are only available for access through HiPay Mobile support.

Parameter	Description	Example
code	General purpose field	123
message	General purpose field	OK

## Sample Response

### URL notification Example

---

```
<?xml version="1.0" encoding="UTF-8"?>
<response status="1">
  <code>123</code>
  <message>OK</message>
</response>
```

---

# Signature verification

## Check the signature of a response from the API

**Description** The response to a signed request is also signed. This signature is returned in an HTTP header named “X-Allopass-Response-Signature”.

**Example**

X-Allopass-Response-Signature:  
40b9f7897452a0d4494e42025b10fe31001f1f83

**Procedure** To verify the signature, proceed as follow:

Steps	Action
1	<p>Get the body of the response (XML or JSON).</p> <pre>&lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;response xmlns="https://api.allopass.com/rest" code="0" message="OK"&gt; &lt;id&gt;123456&lt;/id&gt; &lt;name&gt;&lt;![CDATA[PRODUCT NAME]]&gt;&lt;/name&gt; &lt;purchase_url&gt;&lt;![CDATA[http://localhost/purchase]]&gt;&lt;/purcha se_url&gt; &lt;forward_url&gt;&lt;![CDATA[http://localhost/product]]&gt;&lt;/forward_ url&gt; &lt;/response&gt;</pre>
2	<p>Create a string by concatenating the body of the response and the secret key.</p> <pre>&lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;response xmlns="https://api.allopass.com/rest" code="0" message="OK"&gt; &lt;id&gt;123456&lt;/id&gt; &lt;name&gt;&lt;![CDATA[PRODUCT NAME]]&gt;&lt;/name&gt; &lt;purchase_url&gt;&lt;![CDATA[http://localhost/purchase]]&gt;&lt;/purcha se_url&gt; &lt;forward_url&gt;&lt;![CDATA[http://localhost/product]]&gt;&lt;/forward_ url&gt; &lt;/response&gt;ead9758399359a2bb3b32e240322a11e</pre>
3	<p>Hash the string result with the chosen hash function (in this case: SHA-1)</p> <pre>61434688f14cfdab252f2bf07d14f4ca39d30ff0</pre>
4	<p>Verify that the signature obtained in step 3 is equal to the signature returned in the HTTP header “ X-Allopass-Response-Signature”.</p>

## Example in PHP

```
<?php
/* Using the PHP example script in the category
 * "Merchant Authentication > Make an authenticated request", and assuming
that
 * the URL of the resource request is placed in the "$url" variable */
$url = API_BASE_URL . '/onetime/pricing?' . http_build_query($parameters);

$sock = curl_init($url);
curl_setopt_array($sock, array(
    CURLOPT_HEADER => true,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_FOLLOWLOCATION => false,
    CURLOPT_CONNECTTIMEOUT => 10,
    CURLOPT_LOW_SPEED_TIME => 10,
    CURLOPT_TIMEOUT => 10
));
$response = curl_exec($sock);

if (0 < ($curlErrno = curl_errno($sock))) {
trigger_error("CURL Error ($curlErrno): " . curl_error($sock),
E_USER_NOTICE);
header('Location: /error/unavailable.php');
exit();
}
$httpStatusCode = curl_getinfo($sock, CURLINFO_HTTP_CODE);
$httpHeaderSize = curl_getinfo($sock, CURLINFO_HEADER_SIZE);
curl_close($sock);

// Read the API response returned in the $response variable by the
curl_exec () function

$responseHeaders = array();
$rawHeaders = substr($response, 0, $httpHeaderSize - 4);
$responseBody = substr($response, $httpHeaderSize);

/* Build an associative array from HTTP headers
 * returned by the Allopass API. For example:
 * Content-Type: text/xml
 * X-Allopass-Response-Signature:
61434688f14cfdab252f2bf07d14f4ca39d30ff0
 *
 * becomes:
 * Array(
 * 'Content-Type' => 'text/xml',
 * 'X-Allopass-Response-Signature' =>
'61434688f14cfdab252f2bf07d14f4ca39d30ff0'
 * )
 */
foreach (explode("\r\n", $rawHeaders) as $header) {
list($name, $value) = explode(':', $header);
$responseHeaders[$name] = $value;
}

if (isset($responseHeaders['X-Allopass-Response-Signature'])) {
// STEPS 2 and 3: Calculation of the signature
$returnedResponseSignature = $responseHeaders['X-Allopass-Response-
Signature'];
$computedResponseSignature = hash(API_HASH_FUNCTION, $responseBody .
API_SECRET_KEY);

// STEP 4: Checking the signature returned by the API
if (trim($returnedResponseSignature) != trim($computedResponseSignature))
{
header('Location: /error/forbidden.php');
exit();
}
}
}
```

## Check the signature of a notification

### Description

A unique signature is sent (api\_sig) each time that HiPay Mobile contact a merchant page.

### Verification of the signature

To verify this signature, you will need your API Secret Key.

### Find the Secret Key

You will find this key in your HiPay Mobile back office under the « Toolbox » category.



## Signature verification example

### Example in PHP

```
<?php
$parameters = $_GET;

$signature = $parameters['api_sig'];
unset($parameters['api_sig']);
ksort($parameters);

$secretKey      = ''; // fill here with your personal secret key
$string2compute = '';

foreach ($parameters as $name => $value) {
    $string2compute .= $name . $value;
}

// true if OK, false if not
// if you are using md5 instead of sha1 please replace
if (sha1($string2compute . $secretKey) == $signature) {
    $code      = 0;
    $message   = 'OK';
}
else {
    $code      = 1;
    $message   = 'KO';
}

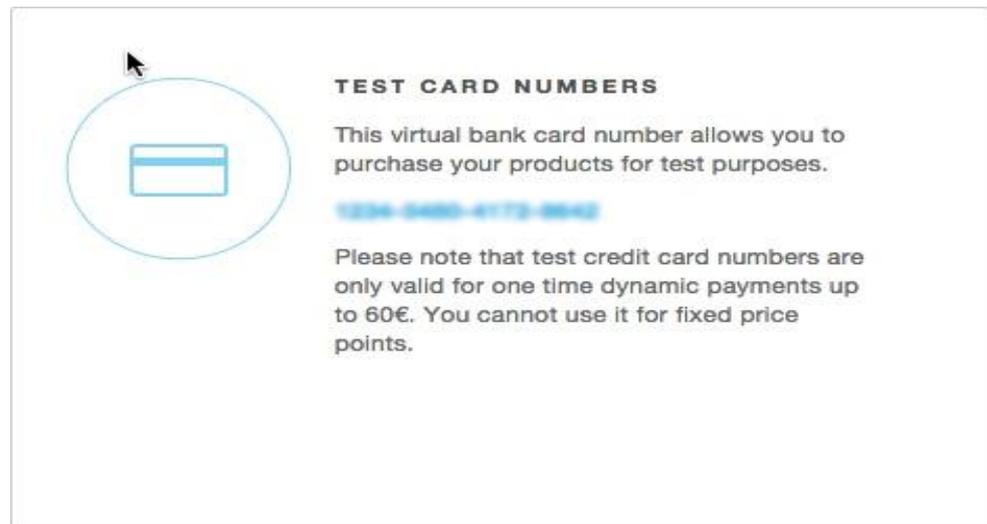
header('Content-Type: text/xml; charset=UTF-8');

echo '<?xml version="1.0" encoding="UTF-8" ?>';
?>
<response status="1">
  <code><?php echo $code; ?></code>
  <message><?php echo $message; ?></message>
</response>
```

# Testing

## Virtual bank card

**Description** HiPay Mobile lets you use a virtual bank card to test your payments; you will find it in your HiPay Mobile back office under the « Toolbox » category.



This virtual bank card number allows you to purchase YOUR products for test purposes

### Note



**Test credit card number is only valid for discrete price points (payments up to 60€). You cannot use it for fixed price points.**

## Test codes

**Description** To facilitate the testing of your integration, for each product set up on the Allopass merchant account, you can specify a test code that will always lead to an unbilled, successful transaction.

**Code security** **It is your responsibility to use strong enough test codes that won't be easily guessed or brute-forced by the end-user.**

### Note



A merchant already using the Allopass API should be particularly careful in generating a new secret key. New keys are automatically taken into account by the API and any request is refused until the key has been updated on the applications and merchant sites.

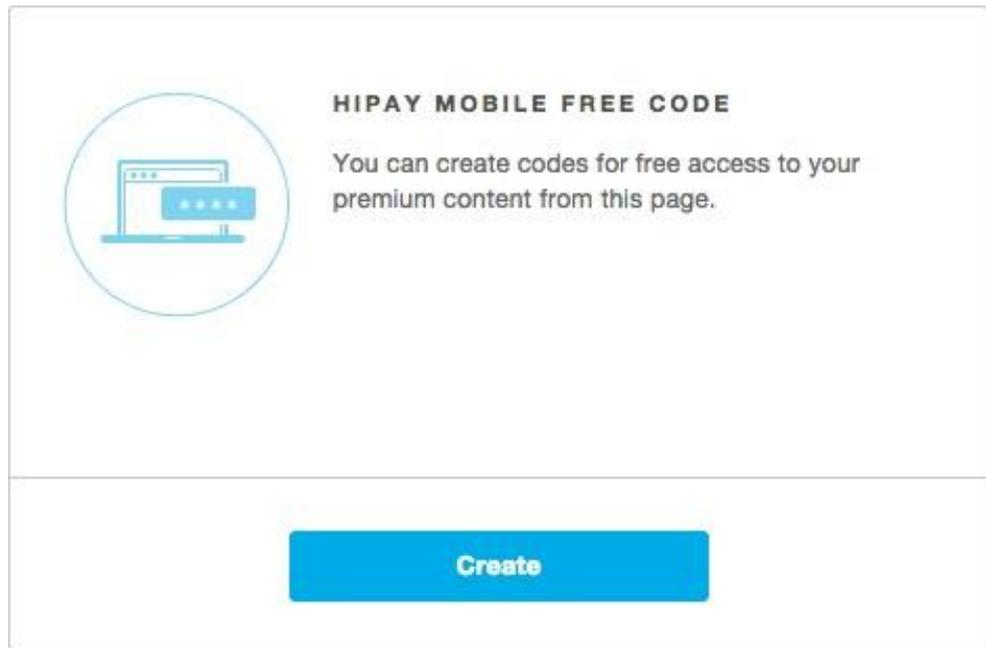
## Free codes

---

**Description** Free codes are more sophisticated than test codes as they are meant to behave more like real codes. Their validity can be adjusted by duration or number of uses, so they can be used for customer support as well.

---

**Free code** You can generate free codes on your HiPay Mobile back office under the « Toolbox » category.



**Note**



Please note that these free codes are tied to specific product IDs. When using the API to generate transactions that are not associated with product IDs, you need to select "*Generic Product*" in the drop-down list.

---

## More questions?

Contact our Merchant support at [contact.mobile@hipay.com](mailto:contact.mobile@hipay.com).