



Integration Guide

InstaPago API version 1.6

Contents

Change Sets	3
Introduction	4
Payment Methods	5
Start	6
Requirements.....	6
Register	7
Keys Creation.....	7
API Use	10
API Entry Point	10
Create Payment	10
Complete Payment	13
Void Payment.....	15
Inquiry Payment	17
Test Cards	19
Response codes API call.....	19
Response code given by the bank.....	20
API Call Examples.....	21
Certification	23

Change Sets

Fecha	Versión	Autor	Descripción
23/08/2013	1.0	John Benavides	○ Creación del Documento
28/10/2013	1.1	Alex Istúriz	○ Nuevo tipo del Request para Payment
07/11/2013	1.2	Alex Istúriz	○ Nuevo parámetro para el Request de Payment ○ Nueva URL del servicio API
20/01/2014	1.3	Alex Istúriz	○ La respuesta del pago incluye el voucher de la compra. Esta debe ser mostrada en el portal y correo electrónico. ○ Recordatorio de generar llave para ambiente " Al Aire
14/03/2014	1.4	Alex Istúriz	○ Nuevo método del servicio API para procesar preautorizaciones
16/05/2014	1.5	Alex Istúriz	○ Modificación al método de preautorizaciones ○ Nuevo método para completar preautorizaciones ○ Nuevo método para anular preautorizaciones ○ Nuevo método para consultar preautorizaciones
20/05/2014	1.5.1	Alex Istúriz	○ Fé de errata URL de completar el pago ○ Ahora todos los métodos devuelven en la respuesta el número de orden si es indicado por el comercio ○ Para el método de crear pago se debe enviar la IP del cliente de manera obligatoria
26/06/2014	1.5.2	César Higuera	○ Nueva imagen de InstaPago en la Guía ○ Correcciones de semántica
29/07/2014	1.5.3	Antonio Romero / Rodrigo Arellano	○ Incorporación de parámetros de certificación ○ Incorporación de recomendaciones
27/03/2015	1.5.4	Yelitza Cedeño / Alex Istúriz	○ Actualización de códigos de ejemplo ○ Actualización de creación de llaves ○ Actualización formato del documento
19/10/2015	1.6	Alex Isturiz	○ Se incluye en todas las respuestas el número de lote asociado a la transacción, el código de respuesta emitido por el banco, monto procesado y fecha de la transacción.

Introduction

InstaPago is a technological solution designed for the electronic commerce market (e-Commerce) in Venezuela and Latin America, with the intention of offering a premium product category, which allows people and companies leverage their expansion capabilities, facilitating mechanisms payment for its customers, with a friendly systems currently using integration.

Any business wishing to receive payments online either from your website or a mobile application, can rely on InstaPago and one of the banks member to achieve their goals.

This Integration Guide is intended for stores and independent developers to integrate their web solutions with the online payment platform InstaPago.

Payment Methods

Credit cards accepted:



Coming Soon

We process through:



Start

One of the main advantages of InstaPago, is the payment by direct communication with our Payment Gateway, either from the web site or a mobile application. The end user or customer is never redirected off-site where are buying, this has the following advantages:

- Better user experience
- Faster buying process
- Greater control of the trade in their processes with customers

Requirements

You should meet the following requirements in order to receive payments through InstaPago:

- A Domain.
- SSL Certificate for the domain.
- Access to source code of the web site.
- Web Hosting

InstaPago is compatible with all programming languages and mobile platforms on the market (PHP, Java, Asp.net C# ó VB, Ruby, Objective-C, etc). Basically, any language that has implemented the HTTP protocol.

Integration with Payment Gateway InstaPago is done through its RESTful API based on HTTP methods.

Register

1. The first step as a developer, is to register on the InstaPago web site and start testing.
2. Go to <https://banesco.instapago.com>
3. Click on "Regístrese" at the end of the form.
4. Complete the requested basic information to create your account. **No matter if it is not a trade, you can create the account with your personal data.**

Keys Creation

By registering on InstaPago, you have direct and immediate access to API Test mode. This will allow you to develop online payment functionality on their web site or mobile application from the time you create your account.

To interact with the API InstaPago, you will receive when registering the lead public connection, which is basically a token that identifies your business and / or user when performing a transaction with InstaPago. Additionally you must create a private key to identify the application that interacts with the API.

To do this, enter the "Opciones" menu and select the "Configuración del API" option as pictured on the next page.

In the section " Configuración del API" you have created the keys shown, and allows you to create new ones. The concept behind this is that a trade can have different outlets. For example, a distributor of electronic products may have a retail outlet or online retail end customers, and another for corporate clients. InstaPago allows you to create a key for each point of payment, so we can consult separate transactions received by both points, and make reconciliations separately.

🏠
🔍
🔄
+

⚙️ Opciones
👤 Ayuda

Tiempo real

2

Transacciones en proceso

Hoy

Transacciones

Mes

Transacciones

Historial de transacciones

35521 Aprobadas 42%

48568 Rechazadas 58%

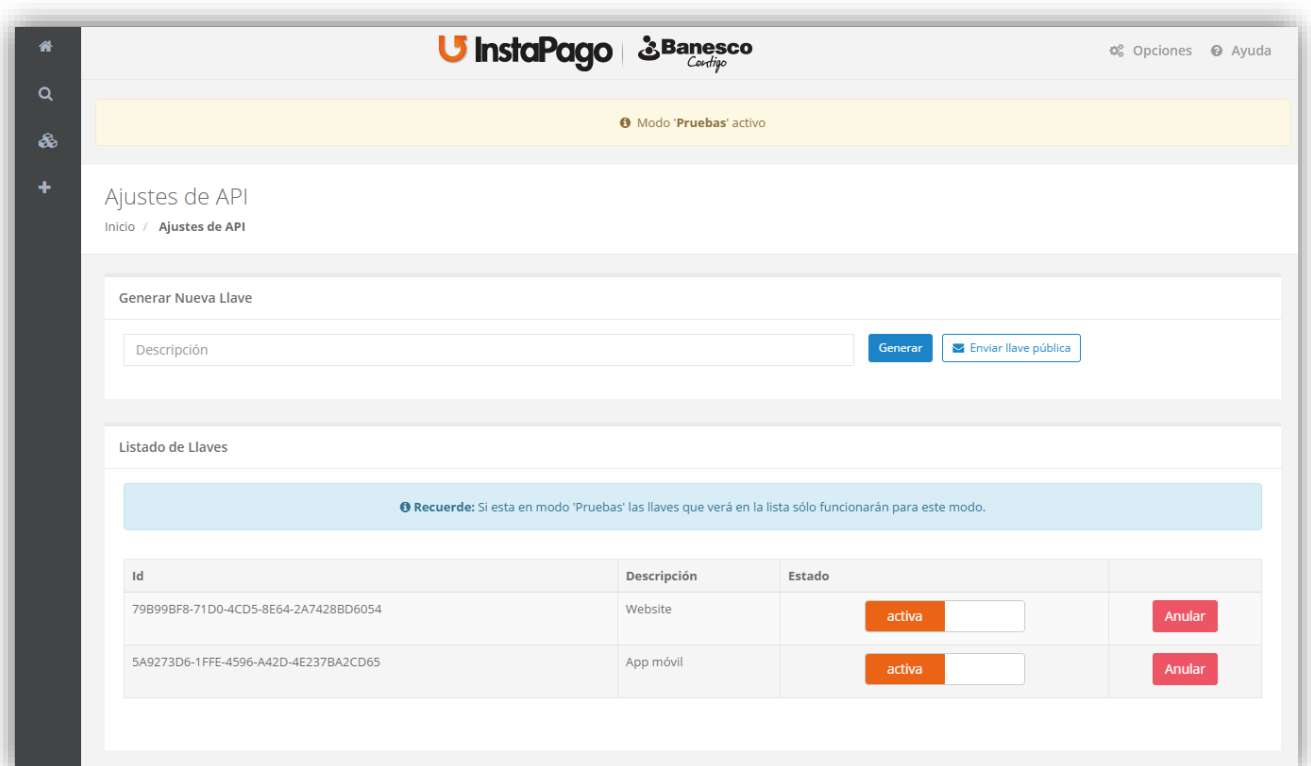
Resumen del Día

<p>Autorizadas</p> <p style="font-size: 24px;">0</p> <p>Transacciones 0 %</p>	<p>Preautorizadas</p> <p style="font-size: 24px;">54</p> <p>Transacciones 1,18 %</p>	<p style="font-size: 24px;">34%</p> <p>Efectivas 1620</p>	<p style="font-size: 24px;">66%</p> <p>Descartadas 3136</p>
<p>Completadas</p> <p style="font-size: 24px;">1007</p> <p>Transacciones 21,16 %</p>	<p>Cerradas</p> <p style="font-size: 24px;">559</p> <p>Transacciones 11,75 %</p>	<div style="border: 2px solid #00a651; border-radius: 50%; padding: 20px; width: 150px; margin: 0 auto;"> <p style="font-size: 24px; font-weight: bold; color: #00a651;">Rechazado</p> <p style="font-size: 24px; color: #00a651;">2,101</p> </div>	
<p>Rechazadas</p> <p style="font-size: 24px;">2101</p> <p>Transacciones 44,16 %</p>	<p>Diferidas</p> <p style="font-size: 24px;">3</p> <p>Transacciones 0,06 %</p>		
<p>Anuladas</p> <p style="font-size: 24px;">1032</p> <p>Transacciones 21,69 %</p>	<p>Fallidas</p> <p style="font-size: 24px;">0</p> <p>Transacciones 0 %</p>		

© 2013 Tecnología InstaPago, C.A. - RIF J-40187810-5

To create a private key, it is assigned a name to identify it (eg "WebSite") and click on the "Generar" button. With this already has a key which can be used to interact with InstaPago in testing mode.

IMPORTANT: After completing the requirements to the bank to access the production environment, you must create keys for this mode again.



The screenshot displays the 'Ajustes de API' (API Settings) page in the InstaPago interface. At the top, the InstaPago and Banesco Contigo logos are visible, along with 'Opciones' and 'Ayuda' links. A yellow banner indicates 'Modo "Pruebas" activo' (Test mode active). The main section is titled 'Ajustes de API' with a breadcrumb 'Inicio / Ajustes de API'. Below this, there is a 'Generar Nueva Llave' (Generate New Key) section with a 'Descripción' (Description) input field, a 'Generar' button, and an 'Enviar llave pública' (Send public key) button. The 'Listado de Llaves' (List of Keys) section features a blue warning box: 'Recuerde: Si esta en modo "Pruebas" las llaves que verá en la lista sólo funcionarán para este modo.' (Remember: If you are in 'Test' mode, the keys you see in the list will only work for this mode). Below the warning is a table with two columns: 'Id' and 'Descripción'. The table contains two rows of keys, both with 'activa' (active) status and 'Anular' (Cancel) buttons.

Id	Descripción	Estado	
79B99BF8-71D0-4CD5-8E64-2A7428BD6054	Website	activa	Anular
5A9273D6-1FFE-4596-A42D-4E237BA2CD65	App móvil	activa	Anular

API Use

API Entry Point

All interactions with InstaPago are made through the following URL:

<https://api.instapago.com>

Create Payment

This method consists of sending data from a credit card payment for authorization to InstaPago:

URL: <https://api.instapago.com/payment>

HTTP Method: POST

Codification: application/x-www-form-urlencoded

Body:

- **KeyId (Required):** Private Key generated from InstaPago website.
- **PublicKeyId (Required):** Key sent by email.
- **Amount (Required):** Payment amount. Ex. 200.00
- **Description (Required):** Payment description.
- **CardHolder (Required):** Card holder name
- **CardHolderID (Required):** Card holder identification number
- **CardNumber (Required):** Card Number (without dashes or spaces)
- **CVC (Required):** Card verification code.

- **ExpirationDate (Required):** Card Expiration date. Format MM/YYYY. Ex.. 10/2014.
- **StatusId (Required):** Type of payment
 - "1": Hold (pre-autorización).
 - "2": Authorize (autorización).
- **IP (Required):** IP address from the client.
- **OrderNumber (Opcional):** Order number of the payment.
- **Address (Opcional):** Billing address associated to the card.
- **City (Opcional):** City from the billing address associated to the card.
- **ZipCode (Opcional):** Zip Code from the billing address associated to the card.
- **State (Opcional):** State from Billing address associated to the card.

Example:

```
POST /Payment HTTP/1.1
Host: https://api.InstaPago.com/payment
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded

KeyId=00000000-0000-0000-0000-000000000000&PublicKeyId=00000000000000000000
000000000000&Amount=10.00&Description=Pago+Servicios+de+InstaPago&CardHolder=
Juan+Perez&CardHolderId=12345678&CardNumber=4111111111111111CVC=123&Expirati
onDate=10%2F2021&StatusId=2&IP=127.0.0.1&OrderNumber=000000010&Address=Ave+L
as+Palmas+Casa+12&City=Ciudad+Bolivar&ZipCode=1902&State=Bolivar
```

Response:

- **Type:** application/json.

- **Values:**

"success": It indicates whether the payment was processed (true o false),

"message": Response description (200 chars max.),

"id": Payment ID (32 chars max.),

"code": Response code (3 chars max.),

"reference": Reference number of the payment (6 chars max.),

"voucher": Transaction ticket,

"ordernumber": Order number of the payment,

"sequence": Bank identification number of the payment (12 chars max.),

"approval": Approval number of the payment (6 chars max.),

"lote": Group number of the transaction (3 chars max.),

"responsecode": Response code given by the bank (2 chars max.),

"deferred": Indicates whether the transaction has been deferred (true o false),

"datetime": Date of the payment (Format MM/dd/yyyy hh:mm:ss tt),

"amount": Amount of the payment (Format 0.00)

Example:

```
{
  "success": true,
  "message": "Pago Aprobado",
  "id": "00000000-0000-0000-0000-000000000000",
  "code": "201",
  "reference": "000003",
  "voucher": "\ \ \ \ &lt;table width=&quot;100%&quot;&gt;\ \ &lt;tbody&gt;\ \
&lt;tr&gt;\ \ &lt;td&gt;\ \ &lt;div style=&quot;border: 1px solid #222; pad
ding: 9px; text-align: center; margin-left: auto; margin-right: auto; margin
-bottom: 10px;&quot; id=&quot;voucher&quot;&gt;\ \ \ \ &lt;style type=&quot;
text/css&quot;&gt;\ \ .normal-left {\ \ font-family: Tahoma;\ \ font-size: 7
pt;\ \ text-align: left;\ \ }\ \ \ \ .normal-right {\ \ font-family: Tahoma;
\ \ font-size: 7pt;\ \ text-align: right;\ \ }\ \ \ \ .big-center {\ \ font-
family: Tahoma;\ \ font-size: 9pt;\ \ text-align: center;\ \ font-weight: 90
0;\ \ }\ \ \ \ .big-center-especial {\ \ font-family: Tahoma;\ \ font-size:
9pt;\ \ text-align: center;\ \ font-weight: 900;\ \ letter-spacing: .9em;\ \
}\ \ \ \ .big-left {\ \ font-family: Tahoma;\ \ font-size: 9pt;\ \ text-ali
gn: left;\ \ font-weight: 900;\ \ }\ \ \ \ ", "sequence": "508600010003", "app
roval": "210220", "lote": "122", "datetime" = "10/25/2015 04:05:11 PM", "amou
nt", "10.00" }
```

Complete Payment

This method works to process a holding payment or pre-authorization in order to carry out the respective charge.

URL: <https://api.instapago.com/complete>

HTTP Method: POST

Codification: application/x-www-form-urlencoded

Body:

- **KeyId (Required):** Private Key generated from InstaPago website.
- **PublicKeyId (Required):** Key sent by email.
- **Amount (Required):** Payment amount. Ex. 200.00.
- **Id (Required):** Payment ID given by InstaPago.

Example:

```
POST /Payment HTTP/1.1
Host: https://api.Instapago.com/complete
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded

KeyId=00000000-0000-0000-0000-000000000000&PublicKeyId=00000000000000000000
000000000000&Id=00000000-0000-0000-0000-000000000000&Amount=10.00
```

Response

- **Type:** application/json.
- **Values:**
 - "success":** It indicates whether the payment was processed (true o false),
 - "message":** Response description (200 chars max.),
 - "id":** Payment ID (32 chars max.),
 - "code":** Response code (3 chars max.),
 - "reference":** Reference number of the payment (6 chars max.),
 - "voucher":** Transaction ticket,
 - "ordernumber":** Order number of the payment,
 - "sequence":** Bank identification number of the payment (12 chars max.),
 - "approval":** Approval number of the payment (6 chars max.),
 - "lote":** Group number of the transaction (3 chars max.),
 - "responsecode":** Response code given by the bank (2 chars max.),
 - "deferred":** Indicates whether the transaction has been deferred (true o false),
 - "datetime":** Date of the payment (Format MM/dd/yyyy hh:mm:ss tt),
 - "amount":** Amount of the payment (Format 0.00)

Example:

```
{
  "success": true,
  "message": "Pago Aprobado",
  "id": "00000000-0000-0000-0000-000000000000",
  "code": "201",
  "reference": "000003",
  "voucher": "<table width=100%><tbody><tr><td><div style=quot;border: 1px solid #222; padding: 9px; text-align: center; margin-left: auto; margin-right: auto; margin-bottom: 10px;quot; id=quot;voucherquot;><style type=quot;text/css"><.normal-left {font-family: Tahoma;font-size: 7pt;text-align: left;}<.normal-right {font-family: Tahoma;font-size: 7pt;text-align: right;}<.big-center {font-family: Tahoma;font-size: 9pt;text-align: center;font-weight: 900;}<.big-center-especial {font-family: Tahoma;font-size: 9pt;text-align: center;font-weight: 900;letter-spacing: .9em;}<.big-left {font-family: Tahoma;font-size: 9pt;text-align: left;font-weight: 900;}<.big-right {font-family: Tahoma;font-size: 9pt;text-align: right;font-weight: 900;}<.normal-center {font-family: Tahoma;font-size: 7pt;text-align: center}</style></td></tr></tbody></table>",
  "sequence": "508600010003", "approval": "210220", "lote": "122", "datetime" = "10/25/2015 04:05:11 PM", "amount", "10.00" }
}
```

Void Payment

This method works to process a cancellation of a payment, either a payment or a hold.

URL: <https://api.InstaPago.com/payment>

HTTP Method: DELETE

Codification: application/x-www-form-urlencoded

Body:

- **KeyId (Required):** Private Key generated from InstaPago website.
- **PublicKeyId (Required):** Key sent by email.
- **Id (Required):** Payment ID given by InstaPago.

Example:

```
DELETE /Payment HTTP/1.1
Host: https://api.InstaPago.com/payment
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
KeyId=00000000-0000-0000-0000-000000000000&PublicKeyId=00000000000000000000
000000000000&Id=00000000000000000000000000000000
```

Response

- **Type:** application/json.
- **Values:**
 - "**success**": It indicates whether the payment was processed (true o false),
 - "**message**": Response description (200 chars max.),
 - "**id**": Payment ID (32 chars max.),
 - "**code**": Response code (3 chars max.),
 - "**reference**": Reference number of the payment (6 chars max.),

- "voucher"**: Transaction ticket,
- "ordernumber"**: Order number of the payment,
- "sequence"**: Universal identification number of the payment (12 chars max.),
- "approval"**: Approval number of the payment (6 chars max.),
- "lote"**: Group number of the transaction (3 chars max.),
- "responsecode"**: Response code given by the bank (2 chars max.),
- "deferred"**: It indicates whether the transaction has been deferred (true or false),
- "datetime"**: Date of the payment (Format MM/dd/yyyy hh:mm:ss tt),
- "amount"**: Amount of the payment (Format 0.00)

Example:

```
{
  "success": true,
  "message": "Pago Anulado",
  "id": "00000000-0000-0000-0000-000000000000",
  "code": "201",
  "reference": "000003",
  "voucher": "\ \ \ \ &lt;table width=&quot;100%&quot;&gt;\ \ &lt;tbody&gt;\ \
&lt;tr&gt;\ \ &lt;td&gt;\ \ &lt;div style=&quot;border: 1px solid #222; padding: 9px; text-align: center; margin-left: auto; margin-right: auto; margin-bottom: 10px;&quot; id=&quot;voucher&quot;&gt;\ \ \ \ &lt;style type=&quot;text/css&quot;&gt;\ \ .normal-left {\ \ font-family: Tahoma;\ \ font-size: 7pt;\ \ text-align: left;\ \ }\ \ \ \ .normal-right {\ \ font-family: Tahoma;\ \ font-size: 7pt;\ \ text-align: right;\ \ }\ \ \ \ .big-center {\ \ font-family: Tahoma;\ \ font-size: 9pt;\ \ text-align: center;\ \ font-weight: 900;\ \ }\ \ \ \ .big-center-especial {\ \ font-family: Tahoma;\ \ font-size: 9pt;\ \ text-align: center;\ \ font-weight: 900;\ \ letter-spacing: .9em;\ \ }\ \ \ \ .big-left {\ \ font-family: Tahoma;\ \ font-size: 9pt;\ \ text-align: left;\ \ font-weight: 900;\ \ }\ \ \ \ .big-right {\ \ font-family: Tahoma;\ \ font-size: 9pt;\ \ text-align: right;\ \ font-weight: 900;\ \ }\ \ \ \ .normal-center {\ \ font-family: Tahoma;\ \ font-size: 7pt;\ \ text-align: center",
  "sequence": "508600010003", "approval": "210220", "lote": "122", "datetime" = "10/25/2015 04:05:11 PM", "amount", "10.00" }
```


Inquiry Payment

This method works for get information for previously generated payment.

URL: <https://api.InstaPago.com/payment>

HTTP Method: GET

Codification: application/x-www-form-urlencoded

QueryString:

- **KeyId (Required):** Private Key generated from InstaPago website.
- **PublicKeyId (Required):** Key sent by email.
- **Id (Required):** Payment ID given by InstaPago.

Example:

```
POST /Payment HTTP/1.1
Host: https://api.InstaPago.com/payment
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded

KeyId=00000000-0000-0000-0000-000000000000&PublicKeyId=00000000000000000000
000000000000&Id=00000000000000000000000000000000
```

Response

- **Type:** application/json.
- **Values:**
 - "**success**": It indicates whether the payment was processed (true o false),
 - "**message**": Response description (200 chars max.),
 - "**id**": Payment ID (32 chars max.),
 - "**code**": Response code (3 chars max.),
 - "**reference**": Reference number of the payment (6 chars max.),
 - "**voucher**": Transaction ticket,
 - "**ordernumber**": Order number of the payment,

"sequence": Universal identification number of the payment (12 chars max.),

"approval": Approval number of the payment (6 chars max.),

"lote": Group number of the transaction (3 chars max.),

"responsecode": Response code given by the bank (2 chars max.),

"deferred": It indicates whether the transaction has been deferred (true or false),

"datetime": Date of the payment (Format MM/dd/yyyy hh:mm:ss tt),

"amount": Amount of the payment (Format 0.00)

Example:

```
{
  "success": true,
  "message": "Pago Aprobado",
  "id": "00000000-0000-0000-0000-000000000000",
  "code": "201",
  "reference": "000003",
  "voucher": "<table width='100%'><tbody><tr><td style='border: 1px solid #222; padding: 9px; text-align: center; margin-left: auto; margin-right: auto; margin-bottom: 10px;' id='voucher'><style type='text/css'>
    .normal-left { font-family: Tahoma; font-size: 7pt; text-align: left; }
    .normal-right { font-family: Tahoma; font-size: 7pt; text-align: right; }
    .big-center { font-family: Tahoma; font-size: 9pt; text-align: center; font-weight: 900; }
    .big-center-especial { font-family: Tahoma; font-size: 9pt; text-align: center; font-weight: 900; letter-spacing: .9em; }
    .big-left { font-family: Tahoma; font-size: 9pt; text-align: left; font-weight: 900; }
    .big-right { font-family: Tahoma; font-size: 9pt; text-align: right; font-weight: 900; }
    .normal-center { font-family: Tahoma; font-size: 7pt; text-align: center; }
  "sequence": "508600010003",
  "approval": "210220"
}
```

Test Cards

For testing InstaPago provides the following cards to test your API integration:

- **Approved Cards:** You can indicate any value for CardHolderID, CVC and Expiration date.
 - **Visa:** 4111 1111 1111 1111
 - **American Express:** 3782 8224 631 0005
 - **MasterCard:** 5105 1051 0510 5100
 - **Sambil:** 8244 0011 0011 0011
 - **Rattan:** 8244 0211 0011 0011
 - **Locatel:** 8244 0411 0011 0011
- **Rejected Cards:** Any valid credit card Visa, MasterCard, American Express, Sambil, Rattan and Locatel will be rejected. **This does not generate any transaction with the Bank.**

Response codes API call

For all transactions under the API, HTTP response codes correspond to the following states:

- **201:** Payment processing, transaction detailed in the response.
- **400:** Failed to validate the data sent (Additionally a character string is returned with the error description).
- **401:** Authentication error, an error occurred with the keys used.
- **403:** Payment declined by the bank.
- **500:** Internal Server Error.
- **503:** Error processing the input parameters. Check the data sent and try again.

If you receive a code different response to those described above they should be taken as HTTP protocol errors.

Response code given by the bank

All API payment responses have a "ResponseCode" value, which corresponds to the answer that throws the bank when the transaction is processed. Possible values are:

Code	Meaning
00	Successful approval
01	Refer to card issuer
02	Refer to card issuer, special condition
03	Invalid merchant or service provider
04	Pickup card
05	Do not honor
06	Error
07	Pickup card, special condition (other than lost/stolen card)
10	Partial Approval
11	V.I.P. approval
12	Invalid transaction
13	Invalid amount (currency conversion field overflow) or amount exceeds maximum for card program
14	Invalid account number (no such number)
15	No such issuer
19	Re-enter transaction
21	No action taken (unable to back out prior transaction)
25	Unable to locate record in file, or account number is missing from the inquiry
28	File is temporarily unavailable
41	Pickup card (lost card)
43	Pickup card (stolen card)
51	Insufficient funds
52	No checking account
53	No savings account
54	Expired card
55	Incorrect PIN
57	Transaction not permitted to cardholder

58	Transaction not allowed at terminal
59	Suspected fraud
61	Activity amount limit exceeded
62	Restricted card (for example, in Country Exclusion table)
63	Security violation
65	Activity count limit exceeded
75	Allowable number of PIN-entry tries exceeded
76	Unable to locate previous message (no match on Retrieval Reference number)
77	Previous message located for a repeat or reversal, but repeat or reversal data are inconsistent with original message
78	'Blocked, first used'—The transaction is from a new cardholder, and the card has not been properly unblocked.
80	Visa transactions: credit issuer unavailable. Private label and check acceptance: Invalid date
81	PIN cryptographic error found (error found by VIC security module during PIN decryption)
82	Negative CAM, dCVV, iCVV, or CVV results
83	Unable to verify PIN
85	No reason to decline a request for account number verification, address verification, CVV2 verification, or a credit voucher or merchandise return
91	Issuer unavailable or switch inoperative (STIP not applicable or available for this transaction)
92	Destination cannot be found for routing
93	Transaction cannot be completed, violation of law
96	System malfunction, System malfunction or certain field error conditions

API Call Examples

Here are some examples of how to make the API calls for different programming languages. It should be noted that **all calls made to the API InstaPago must be done from the server where the site is hosted or application of trade and not directly from the client device.**

PHP

```
<?php

$url = 'https://api.instapago.com/payment';
$fields = array("KeyId" => " ", "PublicKeyId" => " ", "Amount" => "1", "Description" => "
", "CardHolder" => " ", "CardHolderId" => " ", "CardNumber" => " ", "CVC" => " ",
"ExpirationDate" => " ", "StatusId" => " ", "Address" => " ", "City" => " ", "ZipCode" => "
", "State" => " " );

$ch = curl_init();

curl_setopt($ch, CURLOPT_URL, $url );
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($fields));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

$server_output = curl_exec ($ch);
curl_close ($ch);
echo $server_output ;

?>
```

C#

```

WebRequest __webrequest = WebRequest.Create("https://api.instapago.com/payment") as
HttpWebRequest;

string PostData =
String.Format("KeyId={Account_Key}&&Amount={Monto}&Description=={Texto-
descripcion}&CardHolder={tarjeta-habiente}& CardHolderId={cedula-RIF-tarjeta-
habiente}&CardNumber={numero-tarjeta}&CVC={codigo-seguridad-tarjeta}&ExpirationDate={
fecha-expiracion(mm/yyyy)}&StatusId= {status(1=retenido,2=pagado)",

        formData.Payments.First().Cardholder,
        formData.Payments.First().CardNumber,
        formData.Payments.First().CardNumberId,
        formData.Payments.First().SecurityCode,
        formData.Payments.First().ExpMonth.ToString("00"),
        formData.Payments.First().ExpYear.ToString()
    );
byte[] __postDataStream = Encoding.UTF8.GetBytes(PostData);

__webrequest.Method = "POST";
__webrequest.ContentType = "application/x-www-form-urlencoded";
__webrequest.ContentLength = __postDataStream.Length;
Stream __requestStream = __webrequest.GetRequestStream();
__requestStream.Write(__postDataStream, 0, __postDataStream.Length);
__requestStream.Close();

// response
WebResponse __webresponse = __webrequest.GetResponse();
Stream dataStream = __webresponse.GetResponseStream();
StreamReader reader = new StreamReader(dataStream);
string responseFromServer = reader.ReadToEnd();

JavaScriptSerializer __serializer = new JavaScriptSerializer();
Models.InstapagoRS __instaRs =
__serializer.Deserialize(responseFromServer, typeof(Models.InstapagoRS)) as
Models.InstapagoRS;

if (__instaRs.success == true)
    String __codigoAprobacion = __instaRs.id.ToString();
else
    String __errorMessage = "Lo sentimos, no hemos podido procesar su tarjeta de crédito. El
mensaje del banco fue: " + __instaRs.message;

```

Certification

Certification parameters presented below define the points to be evaluated by the technical team InstaPago, to ensure that trade and its web (or mobile application) portal has the minimum requirements to be certified. This certification is a prerequisite to allow you goes into production with the payment platform.

Our basic parameters are:

- The portal must require user login and/or registration to access the payment process.
- The security certificate (SSL) must be installed for payment process and for the user registration process. The parameters and tests that will be evaluated in the SSL certificate can be found at:
https://www.ssllabs.com/downloads/SSL_TLS_Deployment_Best_Practices.pdf
- Payment Form:
 - No special characters should be allowed in any of the form fields.
 - You can not store sensitive information fields (card number, expiration date, CVV2 / CVC).
 - Field Validations:
 - Name: You must allow only alphabetic characters including the letter ñ and blank so that the user can enter his name.
 - Card Holder Id: You must allow only numeric characters, limit the amount to at least 6 digits and 8 digits.
 - CardNumber: only numeric characters, limited to 15 digits and at least 16 digits.

- Expiration Date: You must not allow past dates, only valid dates that are greater than the current date.
- CVC: only numeric characters, limited to 3 digits and mask digits.

At the end of the form must include the following: "This transaction will be processed safely thanks to the platform:" under this text should include logos InstaPago (left side) and Banesco (right), both the same height.

Example:

El costo de activar el plan mensual es de BsF. 1.000 (iva incluido)

No. Tarjeta VISA MasterCard

Cod. Seguridad

Vencimiento Mes Año

Nombre en la Tarjeta

Cédula o RIF

Monto BsF. 1.000,00

Esta transacción será procesada de forma segura gracias a la plataforma de

InstaPago Banesco Contigo

- Once the user presses the button to accept or process, to enable the collection process, it must be locked or hidden while waiting response system so that the user can not press more than once.
- Once the response is received from InstaPago, display the information for the customer, including the ticket (voucher) generated by the system.

Example:



If you are ready to certify their website, email us at soporte@instapago.com with the following:

- Store Name.
- RIF.
- URL where certification is performed.
- URL where the SSL certificate is installed.