



CREDITCARDS API

Version 3.1
November 8, 2017



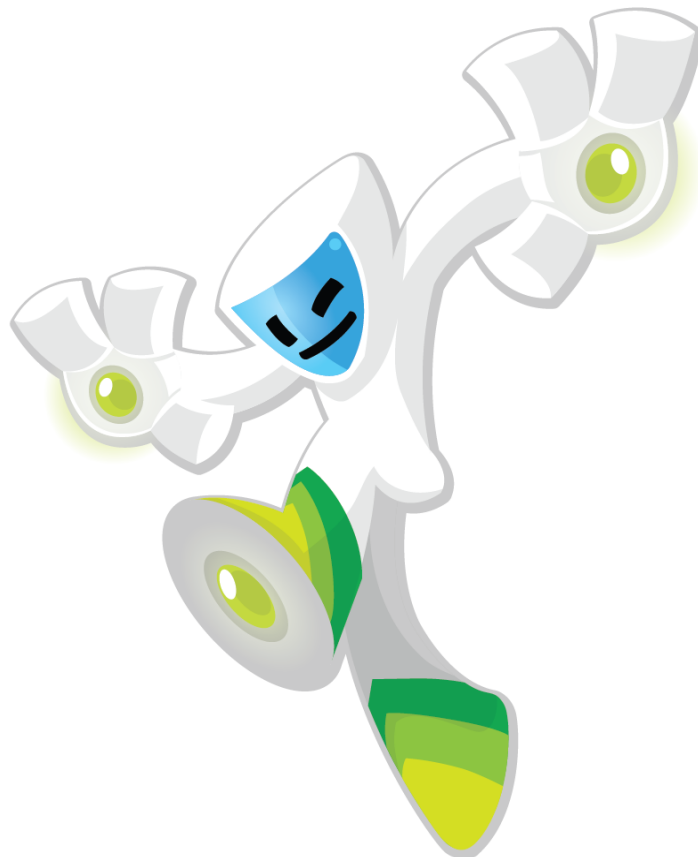
Version History

Date	Changes	Author
September 30, 2013	First version.	
January 2, 2014	Test Credit Card information added.	
January 6, 2014	Removed unused information.	
January 9, 2014	Completed parameters descriptions.	
January 16, 2014	Added authentication.	
March 17, 2014	Debit Card expected parameters list updated.	
March 25, 2014	Mexican Test Credit Card added and <i>auth_token</i> example fixed.	
March 28, 2014	Colombian Test Credit Card changed.	
April 1, 2014	<i>session_id</i> and <i>ip_address</i> parameters added.	
April 22, 2014	Information to obtain the <i>session_id</i> added.	
May 23, 2014	Information about <i>carrier_data</i>	
June 11, 2014	Included <i>vat</i> parameter to debit card.	
July 2, 2014	Include <i>display_name</i> parameter for add card.	
July 31, 2014	Include codes for <i>status_detail</i> .	
August 17, 2014	Unifying documentation	
August 18, 2014	Adding notifications	
September 11, 2014	Adding error messages	
September 23, 2014	Changed heroku URL	
January 28, 2015	Added buyer errors	
February 19, 2015	Added verify method	
December 1, 2015	Added card bin on response for list cards	Abimael Quijada
December 4, 2015	Refund added	Abimael Quijada
December 28, 2015	Added new parameters <i>installments</i> , <i>buyer_fiscal_number</i> and <i>product_discount</i> .	Abimael Quijada
March 7, 2016	Updating info about verify method and status details.	Abimael Quijada
May 11, 2016	Comment about <i>buyer_fiscal_number</i>	Abimael Quijada
Jul 22, 2016	<i>buyer_phone</i> added for add and debit	Abimael Quijada
August 8, 2016	Updating verify method	Abimael Quijada
October 19, 2016	Added frame for do payment	Abimael Quijada

February 7, 2017	Added new parameters <i>installments_type</i> , <i>taxable_amount</i> and <i>tax_percentage</i> . Updating info about <i>product_amount</i> .	Adan Garcia
February 23, 2017	Parameter <i>phone_number</i> deprecated and comment about shipping address.	Citlali Calderón
June 2, 2017	Details about <i>installments_type</i> and <i>installments</i> .	Citlali Calderón
July 4, 2017	Clarification about stoken (callback)	Citlali Calderón
August 25, 2017	Several changes, new add card method, <i>session_id</i> section, <i>authorization_code</i> in callback.	Citlali Calderón
November 8, 2017	New options for <i>installments_type</i> , and new parameters in callback for datafast.	Citlali Calderón

Table of Contents

1. Requirements	5
2. Methods	5
2.1 Add Card	6
2.1.1 Add Card- cookies	6
2.1.2 Add Card- redirect	7
2.2 List Cards	9
2.3 Debit Card	10
2.4 Delete Card	13
2.5 Verify	14
2.6 Refund	16
2.7 Debit card frame	17
3. Obtaining Session Ids	19
4. Testing	19
5. Handling errors	20
6. Receiving transactions	21
6.1 Making your callback	21
6.2 Testing your callback	22
7. Glossary	23



1. Requirements

To start the integration you will need to request to Paymentez Team integrations@paymentez.com for a Development/Sandbox account. Please send us your e-mail to identify you as a developer and the name of your company. We suggest you have an integration e-mail i.e. paymentez@yourcompanydomain.com, if it is not possible any e-mail would be fine.

We will create an “*Application*” and give you the application code. From now this will be the identifier for your Application in the whole integration. We also give you a “*developer*” account based in the e-mail you provided us. We will send you the password via e-mail to access to your developer account. You can access to this configuration here:

Environment	URL
development	https://paymentez.herokuapp.com/
production	https://secure.paymentez.com/

Table 1. Paymentez Developer Console Environments

You can change you account password or if you forget it you can always use the ‘forgot password’ option to recover it. In the Paymentez admin system you will see your transactions, application settings (including application URLs and application key) and so more.

Configurations have to be done for the application in development environment and production environment, URLs and application key are different for every environment. Development environment will be always available for tests even after launching your application to production.

2. Methods

In order to use the API, you need to use one of the following base URLs depending of the environment:

Environment	URL
development	https://ccapi-stg.paymentez.com
production	https://ccapi.paymentez.com

Table 2. API Environments

2.1 Add Card

To add a card to our system, use the following endpoint using a browser:

```
GET /api/cc/add/
```

Figure 1. Add Card Endpoint

There are two options, the first obtaining the result throw the cookies, the second throw the redirection to a success url or failure url.

2.1.1 Add Card- cookies

The parameters expected are described below:

Parameter	Type	Description
application_code	Required	Application identifier (provided by <i>Paymentez</i>). Type: String. Format: Alphanumeric.
uid	Required	User identifier. This is the identifier of the user, you use inside your application; you will receive it in notifications.
email	Required	Email of the user initiating the purchase. Type: String. Format: Valid e-mail format.
session_id	Required	Fraud related parameter. Please see section 3. Type: String Format: 32-length numeric hash.
auth_timestamp	Required	Epoch number used to compute token to verify this request. Type: Integer Format: Epoch number
auth_token	Required	This token will allow us to verify the data integrity. It's a SHA256 made of ALL the parameters, sorted by key and utf-8 <i>urlencoded</i> , appending the <i>auth_timestamp</i> and the <i>Application Key</i> (provided by <i>Paymentez</i>) at the end. Type: String Format: Valid signature. Example in python: <pre> string = "application_code=foo&email=awesome%40user.com &session_id=1234&uid=1&1394829530&ThisI5myK3Y" hashlib.sha256(string).hexdigest() The token result should be: 'f167b243d5abde1b13bc107b0ce4aaf51f 4008939030a29d4a4db1a547101724' </pre>
buyer_phone	Optional	Phone of the user. Type: String. Format: Alphanumeric

Table 3. Description of the parameters required by the Add Card Endpoint (cookies)

An URL example to add card:

https://ccapi-stg.paymentez.com/api/cc/add/?auth_timestamp=1483731456&uid=4&email=ccalderon%40paymentez.com&session_id=Pm9A1rh8slHnGoJ4fyhdOmPkZNdWsYs&application_code=CiColApp&auth_token=acab6a2a1e4be12264de61513118ef98d64a3cba71f32410719df3f0810e6fe4

The result will be settled down in the followings cookies:

pmntz_error_message	
pmntz_add_success	true

Figure 2. Cookies

If an error happened a cookie named *pmntz_error_message* is set with an error message, if our fraud system rejected it, you should see an error like the following:

```
{"verify_transaction": "RB-876"}
```

In that case please go to the Verification section, see the Glossary for more information.

2.1.2 Add Card- redirect

The parameters expected are described below:

Parameter	Type	Description
application_code	Required	Application identifier (provided by <i>Paymentez</i>). Type: String. Format: Alphanumeric.
uid	Required	User identifier. This is the identifier of the user, you use inside your application; you will receive it in notifications.
email	Required	Email of the user initiating the purchase. Type: String. Format: Valid e-mail format.
session_id	Required	Fraud related parameter. Please see section 3. Type: String Format: 32-length numeric hash.
auth_timestamp	Required	Epoch number used to compute token to verify this request. Type: Integer Format: Epoch number
auth_token	Required	This token will allow us to verify the data integrity. It's a SHA256 made of ALL the parameters, sorted by key and utf-8 <i>urlencoded</i> , appending the <i>auth_timestamp</i> and the <i>Application Key</i> (provided by <i>Paymentez</i>) at the end. Type: String Format: Valid signature. Example in python: <pre>string = "application_code=foo&email=awesome%40user.com&session_id=1234&uid=1&1394829530&ThisI5myK3Y"</pre>

		<pre>hashlib.sha256(string).hexdigest()</pre> <p>The token result should be: 'f167b243d5abde1b13bc107b0ce4aaf51f4008939030a29d4a4db1a547101724'</p>
buyer_phone	Optional	<p>Phone of the user.</p> <p>Type: String. Format: Alphanumeric</p>
response_type	Required	<p>The type of response, must be "redirect"</p> <p>Type: String. Format: "redirect"</p>
success_url	Required	<p>The user will be redirected if the transaction were correct.</p> <p>Type: String Format: URL</p>
failure_url	Required	<p>The user will be redirected if the transaction were wrong.</p> <p>Type: String Format: URL</p>

Table 4. Description of the parameters required by the Add Card Endpoint (redirect)

An example of this type of add card:

https://ccapi-stg.paymentez.com/api/cc/add/?application_code=CiColApp&email=ccalderon%40paymentez.com.com&session_id=acefa93f9c9541418a06c9bc4fc3ab33&uid=1234&auth_timestamp=1490113106066&auth_token=40be43147946939a2403584389d287d8f1ec1cb6e8502d96e96878f61a96a84d&response_type=redirect&success_url=http://google.com&failure_url=https://secure.paymentez.com

2.2 List Cards

To list the cards related to a user, use the following endpoint:

```
GET /api/cc/list/
```

Figure 3. List Cards Endpoint.

The parameters expected are described below:

Parameter	Type	Description
application_code	Required	See Table 3.
uid	Required	See Table 3.
auth_timestamp	Required	See Table 3.
auth_token	Required	See Table 3.

Table 5. Description of the parameters required by the List Cards Endpoint.

The description of the fields present in the response are described below:

Parameter	Description
card_reference	New Card identifier. This code is unique among all cards. Type: String. Format: Long Integer
type	Abbreviated Card type. Type: String Format: "vi" (Visa), "mc" (Mastercard), "ax" (Amex), "di" (Dinners)
name	Card holder name. Type: String. Format: Alphanumeric.
termination	Card last four numbers. Type: String. Format: 4-length string.
expiry_month	Card expiry month. Type: String Format: 2-length string.
expiry_year	Card expiry year. Type: String Format: 4-length string.
bin	Card bin. Type: String Format: 6-length string.

Table 6. Description of the parameters returned by List Cards Endpoint.

2.3 Debit Card

To debit a user saved card use the following endpoint:

```
POST /api/cc/debit/
```

Figure 4. Debit Card Endpoint.

The parameters expected are described below:

Parameter	Type	Description
card_reference	Required	New Card identifier. This code is unique among all cards. Type: String. Format: Long Integer
product_amount	Required	Amount to debit. Type: Number. Format: Decimal with two fraction digits (e.g: 10.00).
product_description	Required	Clear descriptions help players to better understand what they're buying. Type: String. Format: Free.
dev_reference	Required	Merchant order reference. You will identify this purchase using this reference. Type: String. Format: Free.
vat	Required	Sales tax amount, included in product cost. Type: Number. Format: Decimal with two fraction digits (e.g: 10.00).
ip_address	Required	User IP address. Type: String Format: Valid v4 IP address (e.g. "192.0.2.30").
session_id	Required	See Table 3.
application_code	Required	See Table 3.
uid	Required	See Table 3.
email	Required	See Table 3.
auth_timestamp	Required	See Table 3.
auth_token	Required	See Table 3.
product_discount	Optional	Amount to be discounted. This field is informative only, doesn't affect the final amount. Type: Number. Format: Decimal with two fraction digits (e.g: 10.00).
Installments	Optional	The number of installments for the payment, only for COP, BRL and USD (Datafast). Type: Number in interval 1 to 24.
buyer_fiscal_number	Required	The fiscal number given by the buyer(is not included at the auth_token). Type: String. Format: Free.

buyer_phone	Optional	Phone of the user. Type: String. Format: Alphanumeric
installments_type	Optional	Only available for Datafast (Ecuador). The type of installment payment. Type: Number. Valid values: 0 – Revolving credit (<i>rotativo</i>). 1 – Revolving and deferred without interest (The bank will pay to the commerce the installment, month by month). 2 – Deferred with interest. 3 – Deferred without interest. 7 – Deferred with interest and months of grace. 9 – Deferred without interest and months of grace. 21 – For Diners Club exclusive, deferred with and without interest. 22 – For Diners Club exclusive, deferred with and without interest. Note for types 7 and 9: The number of months of grace are fixed between the bank and the commerce.
taxable_amount	Optional	Only available for Datafast (Ecuador). The taxable amount, if it is zero, it is calculated on the total, Type: Number. Format: Decimal with two fraction digits (e.g: 10.00).
tax_percentage	Optional	Only available for Datafast (Ecuador). The tax percentage. Type: Number. Format: Decimal with two fraction digits (e.g: 14.00).

Table 7. Description of the parameters required by the *Debit Card Endpoint*.

If your application sells goods, you can send shipping information to improve and feed our anti-fraud system detection. If any of the following parameters is included in the request, have to be considered in the creation of the `auth_token`. The next table describe optional parameters :

Parameter	Type	Description
seller_id	Optional	If your application sells a service (i.e. a taxi driver in a taxi application) and you want to identify it on the acquirer's system as a payment reference. Type: String. Format: 18-length Alphanumeric
shipping_street	Optional	The street of the shipping address:

		Type: String. Example: Av Jacutinga
shipping_house_number	Optional	The house number of the shipping address: Type: String. Example: 607
shipping_city	Optional	City for the shipping address: Type: String. Example: São Paulo
shipping_zip	Optional	ZIP code for the shipping address: Type: String. Format: AR, CL: 3 digits + hyphen + 4 digits (e.g. 999-9999) BR: 5 digits + hyphen + 3 digits(e.g. 99999-999) CO, MX: 5 digits (e.g. 99999) VE: 4 digits (e.g. 9999)
shipping_state	Optional	The state of the shipping address. Type: String. Example: SP
shipping_country	Optional	Country for the shipping address. Type: String. Example: ISO-639 two digit country code (e.g. CO, CL, AR, BR, MX)
shipping_district	Optional	District for the shipping address. Type: String. Example: Free
shipping_additional_address_info	Optional	Additional info for the shipping address. Type: String. Example: Free

Table 8. Description of optional parameters used by *Debit Card Endpoint*.

The description of the fields present in the response are described below:

Parameter	Description
transaction_id	Transaction identifier. This is code is unique among all transactions. Type: String. Format: A 36-character string.
status	Operation result. Type: String Format: "success" or "failure"
status_detail	Additional information about the transaction status, see the Glossary for more information. Type: String.

	Format: Free.
payment_date	UTC approval date. Type: String. Format: YYYY-MM-DD HH:MM:SS
amount	Authorized amount. Type: Number Format: Decimal with two fraction digits (e.g: 10.00).
carrier_data	Any extra data relevant to the carrier. Not present if no extra data. Type: String Format: JSON with value/key pairs.
card_data	Extra data for the used creditcard. Type: JSON Format: Key/value pairs. number: last 4 digits; type; quotas: number of payments; account_type: "C" or "D".

Table 9. Description of the parameters returned by *Debit Card Endpoint*.

2.4 Delete Card

To delete a user saved card use the following endpoint:

```
POST /api/cc/delete/
```

Figure 5. *Delete Card Endpoint*.

The parameters expected are described below:

Parameter	Type	Description
card_reference	Required	See Table 5.
application_code	Required	See Table 3.
uid	Required	See Table 3.
auth_timestamp	Required	See Table 3.
auth_token	Required	See Table 3.

Table 10. Description of the parameters required by the *Delete Card Endpoint*.

The response will be an HTTP Status 200 OK.

2.5 Verify

Sometimes an add card or debit transaction would need to be verified with a code from the financial entity that charges the card. When the buyer gets the verification code from his bank, you can verify the operation making a request to:

```
POST /api/cc/verify/
```

Figure 6. Verify Endpoint.

The parameters expected are described below:

Parameter	Type	Description
application_code	Required	See Table 3.
uid	Required	See Table 3.
auth_timestamp	Required	See Table 3.
auth_token	Required	See Table 3.
transaction_id	Required	See Table 8.
type	Required	It identifies if the value is authorization code or amount Type: String Format: Valid strings: <ul style="list-style-type: none"> • BY_AMOUNT • BY_AUTH_CODE
value	Required	The authorization code provided by the financial entity to the buyer or the transaction amount. Type: String

Table 11. Description of the parameters required by the *Verify Endpoint*.

Python Example:

- Parameters:

```
application_code=AbiColApp
uid=123
type=BY_AUTH_CODE
value=003361
transaction_id=CB-5819
auth_timestamp=2233
```
- Generating the auth_token:

```
application_key = "2PmoFfjZjzjKTnuSYCFySMfHIOIBz7"
plain_text = "application_code=AbiColApp&transaction_id=CB-5819&type=BY_AUTH_CODE&uid=123&value=003361&2233&2PmoFfjZjzjKTnuSYCFySMfHIOIBz7"
token = hashlib.sha256(plain_text).hexdigest()
(token is "847cd4db505ab1203b9e78efe1ba21fc552242f303f274770bd19fe0c3a91b14")
```
- Request parameters:

```
application_code=AbiColApp&uid=123&type=BY_AUTH_CODE&value=003361&transaction_id=CB-5819&auth_timestamp=2233&auth_token=fe575f683095b9f94021d5bb26381977d286d5ee0a20539214f2fc4c534de5aa
```

If the verification_code is incorrect you will get an error as the next:

```
{"errors": [{"description": "VerificationError", "details": { "attempts": 1 }, "code": 7 }]}
```

You have 3 attempts before get an error like:

```
{"errors": [{"description": "InvalidTransaction", "details": {}, "code": 7}]}
```

This error also applies when the verification time was exceeded, that time depends for each carrier but the maximum time is 72 hrs.

Finally if the verification_code is correct the result will be:

```
{
  "status": 1,
  "payment_date": "2016-03-07T17:22:00",
  "amount": 100,
  "transaction_id": "CB-5819",
  "status_detail": 3
}
```

If the transaction is from an add card the card now will be available for the user, otherwise the transaction will be announce and notified to the user.

2.6 Refund

Use to refund approved transactions.

POST /api/cc/refund

Figure 7. Refund Endpoint

The parameters expected are described below:

Parameter	Type	Description
application_code	Required	Is the code associated to the application created by Paymentez Team, if you don't have it, please ask your contact on Paymentez Team for it. Type: String. Format: Alphanumeric.
transaction_id	Required	Transaction identifier returned on <i>Debit Card</i> Type: String. Format: Alphanumeric.
auth_timestamp	Required	Epoch number used to compute token to verify this request. Type: Integer Format: Epoch number
auth_token	Required	This token will allow us to verify the data integrity. Please see section 4.1 Type: String Format: Valid signature.

Table 12. Parameters used by refund.

The description of the fields present in the response (JSON format) are described below:

Parameter	Description
status	Operation result. Type: String Format: <i>success</i> if the refund was approved by the carrier, <i>pending</i> because some carriers could take more than 10 days for approval and <i>failure</i> if an error happened.
detail	Additional information about the transaction status. Type: String. Format: Free.

Table 13. Parameters returned on refund endpoint.

2.7 Debit card frame

This is used when you want to do a purchase with out adding the credit card, show a screen with a form where the user can input his credit card data and pay.

```
GET /api/cc/pay
```

Figure 8. Debit card frame

The parameters expected are described below:

Parameter	Type	Description
application_code	Required	Is the code associated to the application created by Paymentez Team, if you don't have it, please ask your contact on Paymentez Team for it. Type: String. Format: Alphanumeric.
uid	Required	See Table 3.
auth_timestamp	Required	See Table 3.
auth_token	Required	See example below.
dev_reference	Required	Merchant order reference. You will identify this purchase using this reference. Type: String. Format: Free
product_description	Required	Type: String. Format: Free.
product_code	Required	A code that identifies your product. Type: String. Format: Alphanumeric.
product_amount	Required	Amount to debit. Type: Number. Format: Decimal with two fraction digits (e.g: 10.00).
success_url	Required	The user will be redirected if the transaction were correct. Type: String Format: URL
failure_url	Required	The user will be redirected if the transaction were wrong. Type: String Format: URL
review_url	Required	The user will be redirected if the transaction were rejected by antifraud system and needs verification. Type: String Format: URL
installments_type	Optional	Only available for Datafast (Equador). The type of installment payment. Type: Number. Valid values: 0 – Revolving credit (<i>rotativo</i>). 1 – Revolving and deferred without interest (The bank will pay to the commerce the installment, month by month). 2 – Deferred with interest. 3 – Deferred without interest. 7 – Deferred with interest and months of grace. 9 – Deferred without interest and months of grace. 21 – For Diners Club exclusive, deferred with and without interest.

		22 – For Diners Club exclusive, deferred with and without interest. Note for types 7 and 9: The number of months of grace are fixed between the bank and the commerce.
taxable_amount	Optional	Only available for Datafast (Equador). The taxable amount, if it is zero, it is calculated on the total, Type: Number. Format: Decimal with two fraction digits (e.g: 10.00).
tax_percentage	Optional	Only available for Datafast (Equador). The tax percentage. Type: Number. Format: Decimal with two fraction digits (e.g: 14.00).

Table 14. Parameters debit card frame.

Example:

- Parameters:

```

application_code=AbiColApp
uid=tester
auth_timestamp=1476831509
dev_reference=test-tr
product_description=10 pesotes
product_code=prod10
product_amount=10.00
success_url=http://secure.paymentez.com
failure_url=http://google.com
review_url=http://hotmail.com

```

- Plain text with ordered parameters, only those are needed in order to generate the token, plus your application_key in this case 2PmoFfjZjzKTnuSYCFySMfHIOBz7

```

plain_text =
application_code=AbiColApp&dev_reference=test-tr&product_amount=10.00&product_code=prod10&product_description=10
pesotes&uid=tester&1476831509&2PmoFfjZjzKTnuSYCFySMfHIOBz7

```

- URL encoded:

```

plain_text =
application_code=AbiColApp&dev_reference=test-
tr&product_amount=10.00&product_code=prod10&product_description=10+pesotes&uid=tester&1476831509&2PmoFfjZjzKTnu
SYCFySMfHIOBz7

```

- auth_token = hashlib.sha256(plain_text).hexdigest()

```
5267787527883d2e84fe9432041208327483e16349a8057d55b11362dfebd261
```

- Final URL =

```

https://ccapi-stg.paymentez.com/api/cc/pay/?application\_code=AbiColApp&uid=tester&auth\_timestamp=1476831509&auth\_token=5267787527883d2e84fe9432041208327483e16349a8057d55b11362dfebd261&dev\_reference=test-tr&product\_description=10 pesotes&product\_code=prod10
https://ccapi-stg.paymentez.com/api/cc/pay/?application\_code=AbiColApp&uid=tester&auth\_timestamp=1476831509&auth\_token=5267787527883d2e84fe9432041208327483e16349a8057d55b11362dfebd261&dev\_reference=test-tr&product\_description=10 pesotes&product\_code=prod10

```

3. Obtaining Session Ids

The session ID is a parameter Paymentez use for fraud purposes. Obtaining one depends on the platform.

For *Windows Phone* and *browser-based* applications, *Paymentez* provides a *JavaScript* library to do the job. To use it follow the example below:

```
<script type="text/javascript">
  function initSessionID(sessionID) {
    alert("Callback This is the loaded sessionID: " + sessionID);
  }
</script>
<script
  class="paymentez-session-id-js"
  src="[API_URL]/js/paymentez-session-id.js?onload=initSessionID">
</script>
// Replace API URL with the proper URL (see Table 1).
```

Figure 9. *SessionID* JavaScript usage example.

4. Testing

Please use the following credit card to test in our development environment:

Name: [anything you want]
Type: VISA
Number: 4005 5800 0004 0541
CVC: 519
Expiration date: 12/18
Currency: COP

Name: Antonio Flores Aldama
Type: MASTERCARD
Number: 5453 7500 0000 0001
CVC: 000
Expiration date: 10/16
Currency: MXN

5. Handling errors

Below you will find the mainly possible errors that the API can return:

Code	Description	HTTP Code
0	<i>UnauthorizedError</i>	401
1	<i>InsuficientParamsError</i>	403
2	<i>InvalidFormatError</i>	403
3	<i>OperationNotAllowedError</i>	403
4	<i>InvalidConfigurationError</i>	403
7	<i>VerificationError</i>	200
33	<i>BuyerBlockedPaymentError</i>	403
34	<i>BuyerBlockedNotPendingPaymentError</i>	403

Table 15. API errors.

The response will always include the errors and the fields related to each error. See an example below:

```
{
  errors: [
    {code: 2, description: "InvalidFormatError", details: ["name", "url"]},
  ]
}
```

Figure 10. Error example.

6. Receiving transactions

6.1 Making your callback

Every time a transaction gets approved or cancelled you will get an HTTP POST request from Paymentez to your `callback_url` (configured using the admin cpanel). The POST includes the following fields:

- `date`: *Transaction date (used for approved numbers in the Dashboard).*
- `paid_date`: *Transaction paid date (used for approved numbers in the Dashboard).*
- `application_code`: *Application code (This will be always the same).*
- `user_id`: *User id provided by Developer or Social Networks.*
- `transaction_id`: *Transaction identifier from Carrier.*
- `recurrent_transaction_id`: *First transaction (transaction identifier from Carrier) of a recurrency series. This only applies for recurrent transactions.*
- `product_id`: *Product identifier.*
- `token`: *Deprecated Now we use token which is more secure.*
- `stoken`: *MD5 hash of [transaction_id]_[application_code]_[user_id]_[app_key]*

Example(python libraries):

```
transaction_id = 123
app_id = HF
user_id = 123456
app_key = 2GYx7SdjmbucLKE924JVfcmCl8t6nB
for_md5 = "123_HF_123456_2GYx7SdjmbucLKE924JVfcmCl8t6nB"
stoken = hashlib.md5(for_md5).hexdigest()
So the stoken is e242e78ae5f1ed162966f0eacaa0af01
```

- `currency`: *Amount currency (BRL, MXN, ARS, COP, USD, VEF)*
- `gross_value`: *Gross amount in cents*
- `num_coins`: *Amount of coins to deliver (only when the product is present)*
- `product_description`: *Description of purchased product (same as the one in the transaction)*
- `carrier`: *Payment method carrier (see codes below)*
- `payment_method`: *Payment method type (see codes below)*
- `dev_reference`: *Reference used by developer, it has to be unique.*
- `status`: *Transaction status (see values below)*
- `test_mode`: *Sent only when the transaction is a test, the value will be 1.*
- `buyer_first_name`: *The buyer first name, sent if the carrier requires this information.*
- `buyer_last_name`: *The buyer last name, sent if the carrier requires this information.*
- `buyer_phone`: *Buyer's phone, sent if the carrier requires this information.*
- `buyer_ip`: *Buyer's IP.*
- `buyer_email`: *Buyer's email*
- `buyer_street`: *Buyer's street address, sent if the carrier requires this information.*
- `buyer_number`: *Buyer's house number, sent if the carrier requires this information.*
- `buyer_complement`: *Buyer's complement of address, sent if the carrier requires this information.*
- `buyer_district`: *Buyer's district address, sent if the carrier requires this information.*
- `buyer_city`: *Buyer's city, sent if the carrier requires this information.*
- `buyer_state`: *Buyer's state/province, sent if the carrier requires this information.*
- `buyer_zip_code`: *Buyer's zip code, sent if the carrier requires this information.*
- `buyer_country`: *Buyer's country, sent if the carrier requires this information.*
- `pm_user_id`: *Buyer's ID in Paymentez*
- `usd_amount`: *The product amount in dollars.*

- `total_net_value`: *The transaction total net value(only for transactions before 2013/2/1 20:39)*.
- `status_detail`: *Additional information about the status of the transaction(see values below)*.
- `bank_name`: *Bank name if available*.
- `authorization_code`: *The authorization_code sended from carrier (only if approved and for Datafast)*.
- `bin`: *The bin of the credit card (only for Datafast)*.
- `number`: *The last four digits of the credit card (only for Datafast)*.
- `card_type`: *The card type (only for Datafast)*.

Below you can see the values that can take the `status` field:

- **1** Approved
- **2** Cancelled
- **4** Rejected

For every transaction you must return an HTTP status, this status is only used to know that you received correctly the call:

- **200** success
- **201** product_id error
- **202** user_id error
- **203** token error
- **204** transaction_id already received

The `product_id` may be set through Paymentez control panel. The `token` is an important field we send for you to confirm the information authenticity. You just need to generate and match the token against the one you receive to be sure that the POST came from Paymentez. If your server doesn't respond with an HTTP 200 OK message, the POST will be retried until get and HTTP 204 status. You must store this information from all transactions in your database and always check the `transaction_id` to make sure you are not getting a duplicated POST.

Additionally to approve transactions we also send you those approved transactions that get **cancelled**, this time the only difference is the **status** value, which will be **2**. In this case you should answer with 204 (so we don't send it again) and should update the transaction status so you ensure your data and accounting matches with Paymentez.

Once you have your callback URL you can go to our Control Panel and configure it using the tab "Applications".

6.2 Testing your callback

You can test your callback using our Control Panel. You just need to go to "Applications" and place the mouse over one of them and you will see the option.

7. Glossary

Dates

All dates exposed in the API are based on UTC (unless a time zone is shown).

Countries

By now the API is only enabled for Colombia, Ecuador, Brazil and Mexico.

Status Detail

Below you can see the values for the status_detail parameter.

Code	Description
1	Verification required, please see Verification section.
3	Paid.
6	Fraud.
7	Refund.
8	Chargeback
9	Rejected by carrier.
10	System error.
11	Paymentez fraud.
12	Paymentez blacklist.
13	Time tolerance.
19	Invalid Authorization Code.
20	Authorization code expired.
21	Paymentez Fraud - Pending refund
22	Invalid AuthCode - Pending refund
23	AuthCode expired - Pending refund
24	Paymentez Fraud - Refund requested
25	Invalid AuthCode - Refund requested
26	AuthCode expired - Refund requested
27	Merchant - Pending refund
28	Merchant - Refund requested

Table 16. Description of status details

Payment Method

Below you can see the values for the payment_method parameter received on notification.

Code	Description
0	Credit card
1	Boleto (Bank ticket)
3	E – wallet
4	Mobile

Table 17. Description of payment methods

Error messages

Below you can see the error messages:

Message	Description
<i>Error de sistema.</i>	When Paymentez had a problem.

<i>Informacion del comprador inconsistente.</i>	The Buyer information is inconsistent with our system.
<i>Parametros incompletos.</i>	A parameter isn't on the request.
<i>Tu transaccion no pudo ser completada. Fue rechazada por sistema antifraude.</i>	Our fraud detection system found a problem.
<i>Tarjeta rechazada por la entidad financiera. Comunícate con tu banco</i>	The Bank rejected the card.
<i>Tarjeta rechazada por la entidad financiera.</i>	A BIN problem.
<i>Tarjeta rechazada por sistema antifraude.</i>	Our fraud detection system found a problem.

<https://secure.paymentez.com/> Request more information on integrations@paymentez.com