



## Integration Guide

### Transparent Checkout with Split Payment

## Revision History

---

DATE	DESCRIPTION
02/02/2015	Overview
31/03/2015	Including HTTP parameters to split
27/04/2015	Updating URL version and parameters
07/03/2016	Updating the status 7 and 8 nomenclatures
17/07/2017	Updating HTTP parameters and XML elements to split

## Copyright

---

All rights reserved. UOL is a trademark of UNIVERSO ONLINE S/A. UOL's logo is a trademark of UNIVERSO ONLINE S/A. Other trademarks, names, logos and brands are the property of their respective owners. The information contained herein belongs to UNIVERSO ONLINE S/A. All rights reserved. UNIVERSO ONLINE S/A. Address: Avenida Faria Lima, 1384, 6° Floor, São Paulo/SP, ZIP 01452-002, Brazil.

The PagSeguro services are not, nor are intended to be, comparable to financial services offered by financial institutions or credit card companies, and consist simply of a way to enable and monitor the implementation of e-commerce transactions through the management of payments. Any transaction made through PagSeguro is subject to and must comply with the laws of the Federative Republic of Brazil. We recommend that you read the terms and conditions carefully.

## Disclaimer

---

UOL offers no guarantees of any kind [implicit, implied or statutory] with respect to the information contained therein. UOL does not undertake any liability for loss or damage (direct or indirect), caused by error or omission, or resulting from the use of this document or the information contained herein or resulting from the application or use of a product or service described in this document. UOL is entitled to make any changes to any information herein without prior notice.

## Contents

Revision History .....	2
Copyright .....	2
Disclaimer .....	2
Contents.....	3
Overview .....	4
Authentication .....	4
Integration .....	4
Start payment session .....	5
Browser integration .....	5
Obtaining buyer identification .....	6
Obtain the payment method list .....	6
Obtaining credit card brand.....	8
Obtaining credit card token .....	9
Obtaining payment installment options .....	10
Transparent Checkout API with Split Payment .....	12
API call example for Boleto transaction.....	12
API call example for Online Debit .....	14
API call example for Credit Card .....	15
Transparent Checkout API Return .....	16
Parameters List .....	17
Checkout Transparent API Paramenters.....	17
Return Checkout Transparent API Paramenters.....	29
Error list .....	39
Attachments .....	44
API call example for Boleto transaction.....	44
API call example for Online Debit .....	45
API call example for Credit Card .....	47

## Overview

This document describes the integration process of PagSeguro's Transparent Checkout. By integrating your e-commerce system with Transparent Checkout, you can offer all the security and comfort of PagSeguro to your customers without having to leave your website or e-commerce site at the time of purchase. Your website will be able offer payment through Credit Card, Online Debit and "Boleto" (bank payment slip).

## Authentication

All calls to the APIs on the domain `ws.pagseguro.uol.com.br` require authentication. The following HTTP parameters must be sent in all service calls to the aforementioned domain.

PARÂMETRO HTTP	DESCRIÇÃO
appld	ID of Application  <b>Presence:</b> Required. <b>Format:</b> String (< 60 characters). <b>Example:</b> ps_application
appKey	Specifies the token that corresponds to the PagSeguro application that is making the request.  <b>Presence:</b> Required. <b>Type:</b> Text <b>Format:</b> String (32 characters). <b>Example:</b> 861D0B48DBA3485AB2E36675332F2812
authorizationCode	Code that authorizes the application to take action in the name of the seller.  <b>Presence:</b> Required. <b>Type:</b> Text <b>Example:</b> 5A220D39B3B31E8BB4E3EFA4FCC4E6L5

## Integration

To carry out the integration with the Transparent Checkout, you need to follow the steps below:

- ▶ Start a payment session
- ▶ Obtain the payment method list
- ▶ Obtain a credit card banner (applicable only for Credit Card)
- ▶ Obtain a token for the credit card (applicable only for Credit Card)
- ▶ Verify the payment installment options (applicable only for Credit Card)
- ▶ Obtain the buyer's ID
- ▶ Make the payment using the Transparent Checkout API

## Start payment session

To start a Transparent Checkout it is necessary to have a valid session ID. This service returns the session ID that will be used in subsequent JavaScript calls.

### URL:

**POST** <https://ws.pagseguro.uol.com.br/sessions>

### Example:

```
curl https://ws.pagseguro.uol.com.br/sessions/ -d\  
  "appId=ps_application\  
  &appKey=861D0B48DBA3485AB2E36675332F2812\"
```

### Return:

```
1. <?xml version="1.0" encoding="ISO-8859-1"?>  
2. <session>  
3.   <id>620f99e348c24f07877c927b353e49d3</id>  
4. </session>
```

## Browser integration

The Transparent Checkout API has JavaScript functions for a few operations that must be run in the customer's browser. These functions are described below. For these functions, a JavaScript API must be imported at the end of the payment method page:

```
<script type="text/javascript" src=  
"https://stc.pagseguro.uol.com.br/pagseguro/api/v2/checkout/pagseguro.directpayment.js"></script>
```

This JavaScript has an object called **PagSeguroDirectPayment**, which is the interface to access these functions. After importing the file, the **setSessionId** method must be run with the session ID generated previously.

```
<script type="text/javascript">  
  PagSeguroDirectPayment.setSessionId('SessionId');  
</script>
```

In the functions below, the success and error events can occur and JavaScript callback functions can be specified in JSON.

Simply define three JavaScript functions with the name "success," "error," and "complete" in the JSON provided to the method call. The "complete" function will be called regardless of the return and the "success" and "error" functions will be called depending on the outcome. If the return has no error the function called will be "success". If an error occurred then the "error" function will be called.

## Obtaining buyer identification

To use Transparent Checkout, it is necessary to send an identifier of the buyer generated by JavaScript. For this you need to use the `getSenderHash` method. This method has no parameters and returns an identifier. The identifier is **required** for all payment methods.

### Syntax:

```
PagSeguroDirectPayment.getSenderHash();
```



**Attention:** This method has some dependencies and therefore recommend that the `getSender.hash` does not run in `onLoad` of the page. You can run it, for example when the customer clicks the completion payment button.

## Obtain the payment method list

The method `getPaymentMethods` is used to check the PagSeguro's payment method list. The method receive the total amount of the checkout and return a JSON with information like **payment method name**, **exhibition payment method name**, **status (If it was available at the moment)** and the **path for the payment method image** compatible. Check a JSON example below:

### Syntax:

```
1. PagSeguroDirectPayment.getPaymentMethods({
2.     amount: {total checkout amount},
3.     success: {function callback for successful calls},
4.     error: {function callback for unsuccessful calls},
5.     complete: {function callback for completed calls}
6. });
```

### Example:

```
1. PagSeguroDirectPayment.getPaymentMethods({
2.     amount: document.getElementById('amount').value,
3.     success: function(response) {
4.         //get available payment method
5.     },
6.     error: function(response) {
7.         //error handling
8.     },
9.     complete: function(response) {
10.        //optional handling for both result
11.    }
12. });
```

## Return

```

1.  {
2.    "error":false,
3.    "payment.methods":{
4.      "BOLETO":{
5.        "name":"BOLETO",
6.        "options":{
7.          "BOLETO":{
8.            "name":"BOLETO",
9.            "displayName":"Boleto",
10.           "status":"AVAILABLE",
11.           "code":202,
12.           "images":{
13.             "SMALL":{
14.               "size":"SMALL",
15.               "path":"/public/img/payment-methods-flags/42x20/booklet.png"
16.             },
17.             "MEDIUM":{
18.               "size":"MEDIUM",
19.               "path":"/public/img/payment-methods-flags/68x30/booklet.png"
20.             }
21.           }
22.         }
23.       },
24.       "code":2
25.     },
26.     "ONLINE_DEBIT":{
27.       "name":"ONLINE_DEBIT",
28.       "options":{
29.         "BANCO_BRASIL":{
30.           "name":"BANCO_BRASIL",
31.           "displayName":"Banco do Brasil",
32.           "status":"AVAILABLE",
33.           "code":304,
34.           "images":{
35.             "SMALL":{
36.               "size":"SMALL",
37.               "path":"/public/img/payment-methods-flags/42x20/bb.png"
38.             },
39.             "MEDIUM":{
40.               "size":"MEDIUM",
41.               "path":"/public/img/payment-methods-flags/68x30/bb.png"
42.             }
43.           }
44.         },
45.         "code":3
46.       },
47.       "CREDIT_CARD":{
48.         "name":"CREDIT_CARD",
49.         "options":{
50.           "MASTERCARD":{
51.             "name":"MASTERCARD",
52.             "displayName":"MasterCard",
53.             "status":"AVAILABLE",
54.             "code":102,
55.             "images":{
56.               "SMALL":{
57.                 "size":"SMALL",
58.                 "path":"/public/img/payment-methods-flags/42x20/mastercard.png"
59.               },
60.               "MEDIUM":{
61.                 "size":"MEDIUM",
62.                 "path":"/public/img/payment-methods-flags/68x30/mastercard.png"
63.               }
64.             }
65.           }
66.         }
67.       }
68.     }
69.   }


```

```

64.         }
65.     },
66.     },
67.     },
68.     "code":1
69. }}}

```



The payment method images are available in two sizes: **42x20** and **68x30**. You can obtain the  ng the path obtained by the JSON, including the URL <https://stc.pagseguro.uol.com.br> like the example below:

#### 42x20 image example

<https://stc.pagseguro.uol.com.br/public/img/payment-methods-flags/42x20/visa.png>

#### 68x30 image example

<https://stc.pagseguro.uol.com.br/public/img/payment-methods-flags/68x30/visa.png>

## Obtaining credit card brand

This process is required only for the credit card payment method. The **getBrand** method is used to check which card brand is being used. This method takes the BIN of the card (first six digits of the card) and return information like: **brand**, **CVV size**, if **expiration date** is applicable and the **validation algorithm** applies.

#### Syntax:

```

1. PagSeguroDirectPayment.getBrand({
2.     cardBin: {credit card BIN},
3.     success: {function callback for successful calls},
4.     error: {function callback for unsuccessful calls},
5.     complete: {function callback for completed calls}
6. });

```

#### Example:

```

1. PagSeguroDirectPayment.getBrand({
2.     cardBin: document.getElementById('creditCard').value,
3.     success: function(response) {
4.         //brand found
5.     },
6.     error: function(response) {
7.         //error handling
8.     },
9.     complete: function(response) {
10.        //optional handling for both results
11.    }
12. });

```

#### Return:

```

1. {
2.     "brand":{
3.         "name":"visa",

```



```
4.     "bin":411111,  
5.     "cvvSize":3,  
6.     "expirable":true,  
7.     "validationAlgorithm":"LUHN"  
8.   }  
9. }
```

## Obtaining credit card token

This process is required only for the **credit card** payment method. The **createCardToken** method is used to generate the token that will represent the credit card in the call to the Transparent Checkout API. This method receives the following data: card number (**required**), CVV security code (**optional for some cards**), expiration date (**optional for some cards**) and the brand (**optional**).

### Syntax:

```
1. PagSeguroDirectPayment.createCardToken({  
  
2.   cardNumber: {credit card number},  
3.   brand: {brand},  
4.   cvv: {security cvv code},  
5.   expirationMonth: {expiration month (MM)},  
6.   expirationYear: {expiration year (YYYY)},  
7.   success: {function callback for successful calls},  
8.   error: {function callback for unsuccessful calls},  
9.   complete: {function callback for completed calls},  
10. });
```

### Example:

```
1. var param = {  
  
2.   cardNumber: document.getElementById('creditCard').value,  
3.   cvv: document.getElementById('cvv').value,  
4.   expirationMonth: document.getElementById('expirationMonth').value,  
5.   expirationYear: document.getElementById('expirationYear').value,  
6.   success: function(response) {  
7.     //token generated, this must be used in the API call to the Transparent Checkout  
8.   },  
9.   error: function(response) {  
  
10.    //error handling  
  
11.  },  
12.  complete: function(response) {  
13.    //optional handling for both results  
14.  }
```

```
15. }
16.
17. // optional parameter for any call
18. if(document.getElementById('brand').value != '') {
19.     param.brand = document.getElementById('brand').value;
20. }
21.
22. PagSeguroDirectPayment.createCardToken(param);
```

**Return:**

```
1. {
2.     "card":{
3.         "token":"653fe9044cf149f9b7db562431cb130d"
4.     }
5. }
```

## Obtaining payment installment options

This process is required only for the **credit card** payment method. If you wish to show the installment options to the buyer, you must use the **getInstallments** method. This method receives the value to be paid in installments (**required**), the quantity of installments with no interest and the banner to which the installments will apply, returning the configurations of each installment: total value of payment (that must be sent together in the API of the Transparent Checkout), value and amount of installment (that must also be informed in the API of the Transparent Checkout) and an indicator if that installment has interest or not (if the Seller has set up a promotion in PagSeguro). The parameter **maxInstallmentNoInterest** is optional and the value must to be set 2 or above.

If a banner has not been informed as a parameter in the call, the method will return the data for all banners accepted by PagSeguro.

**Syntax:**

```
1. PagSeguroDirectPayment.getInstallments({
2.     amount: {payment amount},
3.     maxInstallmentNoInterest: {no interest installment quantity},
4.     brand: {credit card brand},
5.     success: {function callback for successful calls},
6.     error: {function callback for unsuccessful calls},
7.     complete: {function callback for completed calls},
8. });
```

**Example:**

```
1. PagSeguroDirectPayment.getInstallments({
```

```
2.   amount: document.getElementById('amount').value,
3.   brand: document.getElementById('brand').value,
4.   maxInstallmentNoInterest: 2,
5.   success: function(response) {
6.       //available installment options
7.   },
8.   error: function(response) {
9.       //error handling
10.  },
11.  complete: function(response) {
12.      //optional handling for both results
13.  }
10. });
```

**Return:**

```
1.  {
2.      "error":false,
3.      "installments":{
4.          "visa":[
5.              {
6.                  "quantity":1,
7.                  "totalAmount":16,
8.                  "installmentAmount":16,
9.                  "interestFree":true
10.             },{
11.                 "quantity":2,
12.                 "totalAmount":16.48,
13.                 "installmentAmount":8.24,
14.                 "interestFree":false
15.             }
16.         ]
17.     }
18. }
```

## Transparent Checkout API with Split Payment

This service sends the buyer's data and the payment information to PagSeguro to process the transaction.

### URL:

**POST** <https://ws.pagseguro.uol.com.br/transactions>

**Content-Type:** application/{xml or x-www-form-urlencoded};charset={ISO-8859-1 or UTF-8}

**Accept:** application/vnd.pagseguro.com.br.v3+xml

Every payment must contain one or more items. Each item represents a product or other good that is being purchased. The name of parameters linked to items end with a number. It must to be applied the header

**Accept: application/vnd.pagseguro.com.br.v3+xml** to specify the version 3 of the API.

For example: the parameters **item[1].id**, **item[1].description**, **item[1].amount** and **item[1].quantity** refer to the first item being purchased, while parameters **item[2].id**, **item[2].description**, **item[2].amount** and **item[2].quantity** refer to the second item.

Also you can set one or more sellers to receive part of the payment.

You can split the payment adding the parameter **primaryReceiver** to set the seller who is receiving the transaction, set the parameter **receivers** with one or more receivers and set the transaction amount.

You can define the primaryReceiver and the other receivers by using the seller publicKey, received through the application model.

See below the full example for a request made to PagSeguro's Transparent Checkout API, sending the payment data as HTTP parameters (the lines are separated divided for easier reading).

## API call example for Boleto transaction

```
curl https://ws.sandbox.pagseguro.uol.com.br/transactions/ -d\  
  "appId=seller_app\  
  &appKey=95112EE828D94278BD394E91C4388F20\  
  &payment.mode=default\  
  &payment.method=boleto\  
  &currency=BRL\  
  &item[1].id=0001\  
  &item[1].description=Product 1 Description\  
  &item[1].amount=100.00\  
  &item[1].quantity=1\  
  &notificationURL=https://yourstore.com.br/notification\  
  &reference=ORDER123\  
  &sender.name=Customer Name\  
  &sender.CPF=22111944785\  
  &sender.areaCode=11\  
  &sender.phone=56273440\  
  &sender.email=customer@uol.com.br\  
  &sender.hash=c162e6131dbb9696e8b49147a9c5eacabac0cdca91458ee5177819d0a3c9d565"
```

```
&shipping.address.street=Av. Brig. Faria Lima\  
&shipping.address.number=1384\  
&shipping.address.complement=5o andar\  
&shipping.address.district=Jardim Paulistano\  
&shipping.address.postalCode=01452002\  
&shipping.address.city=Sao Paulo\  
&shipping.address.state=SP\  
&shipping.address.country=BRA  
&shipping.type=3  
&shipping.cost=0.00  
&primaryReceiver.publicKey:PUBCF30546BDF334CC3A59D772E3636D8D3  
&receiver[1].publicKey:PUBCF30546BDF334CC3A59D772E3636D8D4  
&receiver[1].split.amount=20.00  
&receiver[2].publicKey=PUBCF30546BDF334CC3A59D772E3636D8D5  
&receiver[2].split.amount=30.00"
```

The XML parameters are available in attachment of this documentation.

## API call example for Online Debit

```
curl https://ws.sandbox.pagseguro.uol.com.br/transactions/ -d\
  "appId=seller_app\
  &appKey=95112EE828D94278BD394E91C4388F20\
  &payment.mode=default\
  &payment.method=eft\
  &bank.name=itau\
  &currency=BRL\
  &item[1].description=Product 1 Description\
  &item[1].amount=100.00\
  &item[1].quantity=1\
  &notificationURL=https://yourstore.com.br/notification\
  &reference=ORDER123\
  &sender.name=Customer Name\
  &sender.CPF=22111944785\
  &sender.areaCode=11\
  &sender.phone=56273440\
  &sender.email=customer@uol.com.br\
  &sender.hash=c162e6131dbb9696e8b49147a9c5eacabac0cdca91458ee5177819d0a3c9d565\
  &shipping.address.street=Av. Brig. Faria Lima\
  &shipping.address.number=1384\
  &shipping.address.complement=5o andar\
  &shipping.address.district=Jardim Paulistano\
  &shipping.address.postalCode=01452002\
  &shipping.address.city=Sao Paulo\
  &shipping.address.state=SP\
  &shipping.address.country=BRA
  &shipping.type=3
  &shipping.cost=0.00
  &primaryReceiver.publicKey=PUBCF30546BDF334CC3A59D772E3636D8D3
  &receiver[1].publicKey=PUBCF30546BDF334CC3A59D772E3636D8D4
  &receiver[1].split.amount=20.00
  &receiver[2].publicKey=PUBCF30546BDF334CC3A59D772E3636D8D5
  &receiver[2].split.amount=30.00"
```

The XML parameters are available in attachment of this documentation.

## API call example for Credit Card

```
curl https://ws.sandbox.pagseguro.uol.com.br/transactions/ -d\
  "appId=seller_app\
  &appKey=95112EE828D94278BD394E91C4388F20\
  &payment.mode=default\
  &payment.method=creditCard\
  &currency=BRL\
  &item[1].description=Product 1 Description\
  &item[1].amount=100.00\
  &item[1].quantity=1\
  &notificationURL=https://yourstore.com.br/notification\
  &reference=ORDER123\
  &sender.name=Customer Name\
  &sender.CPF=22111944785\
  &sender.areaCode=11\
  &sender.phone=56273440\
  &sender.email=customer@uol.com.br\
  &sender.hash=c162e6131dbb9696e8b49147a9c5eacabac0cdca91458ee5177819d0a3c9d565\
  &shipping.address.street=Av. Brig. Faria Lima\
  &shipping.address.number=1384\
  &shipping.address.complement=5o andar\
  &shipping.address.district=Jardim Paulistano\
  &shipping.address.postalCode=01452002\
  &shipping.address.city=Sao Paulo\
  &shipping.address.state=SP\
  &shipping.address.country=BRA\
  &shipping.type=3\
  &shipping.cost=0.00\
  &installment.quantity=2\
  &installment.value=50.00\
  &installment.noInterestInstallmentQuantity=2\
  &creditCard.token=4as56d4a56d456as456dsa\
  &creditCard.holder.name=Customer Name\
  &creditCard.holder.CPF=22111944785\
  &creditCard.holder.birthDate=27/10/1987\
  &creditCard.holder.areaCode=11\
  &creditCard.holder.phone=56273440\
  &billingAddress.street=Av. Brig. Faria Lima\
  &billingAddress.number=1384\
  &billingAddress.complement=5o andar\
  &billingAddress.district=Jardim Paulistano\
  &billingAddress.postalCode=01452002\
  &billingAddress.city=Sao Paulo\
  &billingAddress.state=SP\
  &billingAddress.country=BRA\
  &primaryReceiver.publicKey=PUBCF30546BDF334CC3A59D772E3636D8D3\
  &receiver[1].publicKey=PUBCF30546BDF334CC3A59D772E3636D8D4\
  &receiver[1].split.amount=20.00\
  &receiver[2].publicKey=PUBCF30546BDF334CC3A59D772E3636D8D5\
  &receiver[2].split.amount=30.00"
```

The XML parameters are available in attachment of this documentation.

## Transparent Checkout API Return

After the call to the Transparent Checkout API, an XML is returned with the transaction data as shown in the example below:

```

1.  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2.  <transaction>
3.      <date>2011-02-05T15:46:12.000-02:00</date>
4.      <lastEventDate>2015-02-02T17:39:14.000-03:00</lastEventDate>
5.      <code>9E884542-81B3-4419-9A75-BCC6FB495EF1</code>
6.      <reference>ORDER123</reference>
7.      <type>1</type>
8.      <status>3</status>
9.      <paymentMethod>
10.         <type>1</type>
11.         <code>101</code>
12.     </paymentMethod>
13.     <paymentLink>
14.         https://pagseguro.uol.com.br/checkout/imprimeBoleto.jhtml?code=314601B208B24A5CA5326000F7BB0D
15.     </paymentLink>
16.     <grossAmount>200.00</grossAmount>
17.     <discountAmount>0.00</discountAmount>
18.     <creditorFees>
19.         <intermediationRateAmount>0.40</intermediationRateAmount>
20.         <intermediationFeeAmount>0.80</intermediationFeeAmount>
21.     </creditorFees>
22.     <netAmount>200.00</netAmount>
23.     <extraAmount>0.00</extraAmount>
24.     <installmentCount>1</installmentCount>
25.     <itemCount>1</itemCount>
26.     <items>
27.         <item>
28.             <id>1</id>
29.             <description>Product 1 Description</description>
30.             <quantity>1</quantity>
31.             <amount>200.00</amount>
32.         </item>
33.     </items>
34.     <sender>
35.         <name>Customer Name</name>
36.         <email>customer@uol.com.br</email>
37.         <phone>
38.             <areaCode>11</areaCode>
39.             <number>56273440</number>
40.         </phone>
41.     </sender>
42.     <shipping>
43.         <address>
44.             <street>Av. Brigadeiro Faria Lima</street>
45.             <number>1384</number>
46.             <complement>1 andar</complement>
47.             <district>Jardim Paulistano</district>
48.             <postalCode>01452002</postalCode>
49.             <city>Sao Paulo</city>
50.             <state>SP</state>
51.             <country>BRA</country>

```



```

52.         </address>
53.         <type>1</type>
54.         <cost>0.00</cost>
55.     </shipping>
56. </transaction>

```

For the payment methods **Boleto** and **Online Debit**, the XML has the item **paymentLink** that returns an URL that gives access to, respectively, the boleto image and the selected payment bank page.



**Warning:** The bank payment page shouldn't be opened into a iFrame.

If an error occurs by a problem in the XML parameters, an error will be returned and it will indicate the errors found in the call. See an example below:

```

1. <errors>

2.     <error>
3.         <code>53031</code>
4.         <message>shipping address city is required.</message>
5.     </error>
6. </errors>

```

In the example above, the call was did without the shipping address city field.

## Parameters List

The table below describes the return parameters of the API of the Transparent Checkout. All parameters are Case sensitive:

### Checkout Transparent API Parameters

PARAMENTER	DESCRIPTION
<b>HTTP Parameter:</b> notificationURL  <b>XML Element:</b> <payment> <notificationURL	<b>URL for the sending of notifications.</b>  <b>Presence:</b> Optional <b>Type:</b> Text <b>Format:</b> A valid URL, restricted to 255 characters
<b>HTTP Parameter:</b> currency  <b>XML Element:</b> <payment> <currency>	<b>Currency used.</b> Indicates the currency in which the payment will be made. At this time, the only option available is <b>BRL</b> (Real).  <b>Presence:</b> Required <b>Type:</b> Text

PARAMETER	DESCRIPTION
	<b>Format:</b> Only the BRL value is accepted
<b>HTTP Parameter:</b> payment.method  <b>XML Element:</b> <payment> <method>	<b>Payment means</b>  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> CreditCard, “boleto” bank payment slip or EFT.
<b>HTTP Parameter:</b> payment.mode  <b>XML Element:</b> <payment> <mode>	<b>Payment method</b>  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> accepts the <b>default</b> option.
<b>HTTP Parameter:</b> item[1].id, item[2].id, etc.  <b>XML Element:</b> <checkout> <items> <item> <id>	<b>Identifier of items.</b> Identifies the items being purchased. You can choose the codes that have meaning in your system and send them in these parameters. PagSeguro does not validate these identifiers, but they cannot repeat within the same payment.  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> Free-form, restricted to 100 characters.
<b>HTTP Parameter:</b> item[1].description, item[2].description, etc.  <b>XML Element:</b> <payment> <items> <item> <description>	<b>Description of items.</b> Describes the items being purchased. The description is the text that PagSeguro displays for each item when the buyer is completing the purchase, therefore, it is important for it to be clear and concise.  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> Free-form, restricted to 100 characters.
<b>HTTP Parameter:</b> item[1].amount, item[2].amount, etc.  <b>XML Element:</b> <checkout> <items> <item> <amount>	<b>Unit value of items.</b> Represents the unit prices of each item being purchased. In addition to being able to contain several items, the transaction can also contain quantities of the same item. This parameter represents the value of a single item unit, that will be multiplied by the quantity to obtain the total value of the line item within the payment.  <b>Presence:</b> Required <b>Type:</b> Number

PARAMETER	DESCRIPTION
	<b>Format:</b> Decimal, with two decimal places separated by a period (ex., 1234,56), greater than 0.00 and less or equal to 9999999.00
<b>HTTP Parameter:</b> item[1].quantity, item[2].quantity, etc.  <b>XML Element:</b> <payment> <items> <item> <quantity>	<b>Quantity of items.</b> Represents the quantity of each item being purchased. In addition to being able to contain several items, the payment can also contain multiple units of the same item. This parameter represents the quantity of an item that will then be multiplied by the value of a single unit to obtain the total value of the line-item within the payment.  <b>Presence:</b> Required <b>Type:</b> Number <b>Format:</b> A whole number greater or equal to 1 and lower or equal to 999.
<b>HTTP Parameter:</b> reference  <b>XML Element:</b> <payment> <reference>	<b>Reference code.</b> Defines a code that refers to the payment. This code is associated with the PagSeguro transaction and is useful to link the transaction in PagSeguro with the a sale in your system.  <b>Presence:</b> Optional <b>Type:</b> Text <b>Format:</b> Free-form, restricted to 200 characters.
<b>HTTP Parameter:</b> extraAmount  <b>XML Element:</b> <payment> <extraAmount>	<b>Additional value to be charged.</b>  <b>Presence:</b> Optional <b>Type:</b> Number <b>Format:</b> Decimal, with two decimal places separated by a period (ex., 1234,56), greater than 0.00 and less or equal to 9999999.00.
<b>XML Element:</b> <payment> <sender>	<b>Buyer's data.</b>
<b>HTTP Parameter:</b> sender.email  <b>XML Element:</b> <payment> <sender> <email>	<b>Buyer's email.</b> Specifies the email of the buyer who is making the payment.  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> A valid email (ex., ), restricted to 60 characters.
<b>HTTP Parameter:</b> sender.name  <b>XML Element:</b> <payment> <sender>	<b>Buyer's full name.</b> Specifies the full name of the buyer who is making the payment.  <b>Presence:</b> Required <b>Type:</b> Text

PARAMENTER	DESCRIPTION
<name>	<b>Format:</b> At least two strings of characters, restricted to a total of 50 characters.
<b>XML Element:</b> <payment> <sender> <documents>	<b>List of buyer's documents (such a ID numbers).</b>
<b>XML Element:</b> <payment> <sender> <documents> <document>	<b>Represents a buyer's document.</b>
<b>XML Element:</b> <payment> <sender> <documents> <document> <type>	<b>Type of buyer's document.</b> Specifies the type of document of the buyer who is making the payment. This field is optional and can be sent if the buyer's data has already been inserted into your system and you wish to prevent him/her from filling out this information again in PagSeguro.  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> Case sensitive. Only the CPF and CNPJ numbers are accepted.
<b>HTTP Parameter:</b> sender.CPF sender.CNPJ  <b>XML Element:</b> <payment> <sender> <documents> <document> <value>	<b>CPF or CNPJ number of buyer</b>  <b>Presence:</b> Required <b>Type:</b> Number
<b>HTTP Parameter:</b> sender.areaCode  <b>XML Element:</b> <payment> <sender> <phone> <areaCode>	<b>Area code (DDD) of buyer.</b> Specifies the area code (DDD) of buyer who is making the payment  <b>Presence:</b> Required <b>Type:</b> Number <b>Format:</b> A 2-digit number that represents a valid area code.

PARAMETER	DESCRIPTION
<b>HTTP Parameter:</b> sender.phone  <b>XML Element:</b> <payment> <sender> <phone> <number>	<b>Telephone number of buyer.</b> Specifies the telephone number of buyer who is making the payment  <b>Presence:</b> Required <b>Type:</b> Number <b>Format:</b> A number comprised of 7 to 9 digits.
<b>HTTP Parameter:</b> sender.hash  <b>XML Element:</b> <payment> <sender> <hash>	<b>Customer ID.</b> Identifier generated by the API JavaScript of PagSeguro that identifies the Customer (hash).  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> Obtained from a JavaScript PagseguroDirectPayment.getSender.hash() call.
<b>XML Element:</b> <payment> <shipping>	<b>Delivery Data</b>
<b>HTTP Parameter:</b> shipping.type  <b>XML Element:</b> <payment> <shipping> <type>	<b>Shipping method of product.</b>  <b>Presence:</b> Optional <b>Type:</b> Number <b>Format:</b> 1-PAC, 2-SEDEX, 3-Unknown
<b>HTTP Parameter:</b> shipping.cost  <b>XML Element:</b> <payment> <shipping> <cost>	<b>Shipping cost</b>  <b>Presence:</b> Optional <b>Type:</b> Number <b>Format:</b> Decimal, with two decimal places separated by a period (ex., 1234.56), greater than 0.00 and less or equal to 9999999.00.
<b>XML Element:</b> <payment> <shipping> <address>	<b>Shipping address data</b>

PARAMETER	DESCRIPTION
<b>HTTP Parameter:</b> shipping.address.country  <b>XML Element:</b> <payment> <shipping> <address> <country>	<b>Country of shipping address</b> Specifies the country of the shipping address of the order.  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> At this time, only the <b>BRA</b> value is allowed.
<b>HTTP Parameter:</b> shipping.address.state  <b>XML Element:</b> <payment> <shipping> <address> <state>	<b>State of shipping address</b> Specifies the state of the shipping address of the product.  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> Two letters, representing the abbreviation of the corresponding Brazilian state.
<b>HTTP Parameter:</b> shipping.address.city  <b>XML Element:</b> <payment> <shipping> <address> <city>	<b>City of shipping address</b> Specifies the city of the shipping address of the product.  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> Free-form. Must be a valid Brazilian city name with at least 2 and no more than 60 characters.
<b>HTTP Parameter:</b> shipping.address.postalCode  <b>XML Element:</b> <payment> <shipping> <address> <postalCode>	<b>Postal code of shipping address</b> Specifies the postal code of the shipping address of the product.  <b>Presence:</b> Required <b>Type:</b> Number <b>Format:</b> An 8-digit number.
<b>HTTP Parameter:</b> shipping.address.district  <b>XML Element:</b> <payment> <shipping> <address> <district>	<b>District of shipping address</b> Specifies the district of the shipping address of the product.  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> Free-form, restricted to 60 characters.

PARAMETER	DESCRIPTION
<b>HTTP Parameter:</b> shipping.address.street  <b>XML Element:</b> <payment> <shipping> <address> <street>	<b>Street name of shipping address</b> Specifies the street name of the shipping address of the product.  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> Free-form, restricted to 80 characters.
<b>HTTP Parameter:</b> shipping.address.number  <b>XML Element:</b> <payment> <shipping> <address> <number>	<b>Number of shipping address.</b> Specifies the number of the shipping address of the product.  <b>Presence:</b> Required <b>Type:</b> Text <b>Format:</b> Free-form, restricted to 20 characters.
<b>HTTP Parameter:</b> shipping.address.complement  <b>XML Element:</b> <payment> <shipping> <address> <complement>	<b>Complement of shipping address.</b> Specifies the complement (block, apartment, etc.) of the shipping address of the product.  <b>Presence:</b> Optional <b>Type:</b> Text <b>Format:</b> Free-form, restricted to 40 characters.
ONLINE DEBIT TRANSACTION DATA	
<b>XML Element:</b> <payment> <bank>	<b>Bank data.</b>
<b>HTTP Parameter:</b> bank.name  <b>XML Element:</b> <payment> <bank> <name>	<b>Bank name</b> Name of bank to which the redirect link will be generated.  <b>Presence:</b> Required for Online Debit <b>Type:</b> Text <b>Format:</b> bradesco, itau, bancodobrasil, banrisul or hsbc

PARAMETER	DESCRIPTION
<b>CREDIT CARD TRANSACTION DATA</b>	
<b>HTTP Parameter:</b> creditCard.token  <b>XML Element:</b> <payment> <creditCard> <token>	<b>Credit Card Token</b> Token returned in the service for obtainment of the credit card token.  <b>Presence:</b> Required for Credit Card <b>Type:</b> Text <b>Format:</b> Unrestricted amount of characters
<b>HTTP Parameter:</b> installment.quantity  <b>XML Element:</b> <payment> <creditCard> <installment> <quantity>	<b>Amount of installments</b> Amount of installments chosen by customer.  <b>Presence:</b> Required for Credit Card <b>Type:</b> Full <b>Format:</b> Whole number
<b>HTTP Parameter:</b> installment.value  <b>XML Element:</b> <payment> <creditCard> <installment> <value>	<b>Value of installments</b> Value of installments obtained in the installment option service.  <b>Presence:</b> Required for Credit Card <b>Type:</b> Number <b>Format:</b> Numeric with 2 decimal places and separated by a period. Ex.: 9.11
<b>HTTP Parameter:</b> installment.noInterestInstallmentQuantity  <b>XML Element:</b> <payment> <creditCard> <installment> <installment.noInterestInstallmentQuantity>	<b>No Interest installment quantity</b> No interest installment quantity offered to the customer. This value should be the same as defined on getInstallments method, on maxInstallmentNoInterest parameter.  <b>Presence:</b> Required if you define a value on maxInstallmentNoInterest parameter from getInstallments method. <b>Type:</b> Numeric <b>Format:</b> Integer. Ex: 10
<b>HTTP Parameter:</b> creditCard.holder.name  <b>XML Element:</b> <payment> <creditCard> <holder> <name>	<b>Name printed on credit card</b>  <b>Presence:</b> Required for Credit Card <b>Type:</b> Text <b>Format:</b> min = 1, max = 50 characters



PARAMETER	DESCRIPTION
<b>HTTP Parameter:</b> creditCard.holder.birthDate  <b>XML Element:</b> <payment> <creditCard> <holder> <birthDate>	<b>Birth date of credit card holder</b>  <b>Presence:</b> Required for Credit Card <b>Type:</b> dd/MM/yyyy <b>Format:</b> 31/12/2013
<b>XML Element:</b> <payment> <creditCard> <holder> <documents>	<b>List of documents of credit card holder.</b>
<b>XML Element:</b> <payment> <creditCard> <holder> <documents> <document>	<b>Represents a document of credit card holder.</b>
<b>XML Element:</b> <payment> <creditCard> <holder> <documents> <document> <type>	<b>Type of document of credit card holder.</b> Specifies the type of document of credit card holder that is being used to make the payment.  <b>Presence:</b> Required for Credit Card <b>Type:</b> Text <b>Format:</b> Case sensitive. Only the <b>CPF</b> number is accepted.
<b>HTTP Parameter:</b> creditCard.holder.CPF  <b>XML Element:</b> <payment> <creditCard> <holder> <documents> <document> <value>	<b>CPF number of credit card holder</b>  <b>Presence:</b> Required for Credit Card <b>Type:</b> Text <b>Format:</b> A valid CPF number
<b>XML Element:</b> <payment> <creditCard> <holder> <phone>	<b>Telephone number of credit card holder.</b>

PARAMETER	DESCRIPTION
<b>HTTP Parameter:</b> creditCard.holder.areaCode  <b>XML Element:</b> <payment> <creditCard> <holder> <phone> <areaCode>	<b>Area code</b>  <b>Presence:</b> Required for Credit Card. <b>Type:</b> Number <b>Format:</b> A 2-digit number that represents a valid area code.
<b>HTTP Parameter:</b> creditCard.holder.phone  <b>XML Element:</b> <payment> <creditCard> <holder> <phone> <number>	<b>Telephone</b>  <b>Presence:</b> Required for Credit Card. <b>Type:</b> Numeric <b>Format:</b> A number comprised of 7 to 9 digits.
BILLING ADDRESS	
<b>XML Element:</b> <payment> <creditCard> <billingAddress>	<b>Billing address.</b>
<b>HTTP Parameter:</b> billingAddress.postalCode  <b>XML Element:</b> <payment> <creditCard> <billingAddress> <postalCode>	<b>Postal code of billing address</b>  <b>Presence:</b> Required for Credit Card. <b>Format:</b> An 8-digit number that represents a valid postal code (ex., 01452002).
<b>HTTP Parameter:</b> billingAddress.street  <b>XML Element:</b> <payment> <creditCard> <billingAddress> <street>	<b>Street Name</b>  <b>Presence:</b> Required for Credit Card. <b>Format:</b> Free-form, restricted to 80 characters.

PARAMETER	DESCRIPTION
<b>HTTP Parameter:</b> billingAddress.number  <b>XML Element:</b> <payment> <creditCard> <billingAddress> <number>	<b>Number</b>  <b>Presence:</b> Required for Credit Card. <b>Format:</b> Free-form, restricted to 20 characters.
<b>HTTP Parameter:</b> billingAddress.complement  <b>XML Element:</b> <payment> <creditCard> <billingAddress> <complement>	<b>Complement</b>  <b>Presence:</b> Required for Credit Card. <b>Format:</b> Free-form, restricted to 40 characters.
<b>HTTP Parameter:</b> billingAddress.district  <b>XML Element:</b> <payment> <creditCard> <billingAddress> <district>	<b>District</b>  <b>Presence:</b> Required for Credit Card. <b>Type:</b> Text <b>Format:</b> Free-form, restricted to 60 characters.
<b>HTTP Parameter:</b> billingAddress.city  <b>XML Element:</b> <payment> <creditCard> <billingAddress> <city>	<b>City</b>  <b>Presence:</b> Required for Credit Card. <b>Format:</b> Must be a valid Brazilian city name with at least 2 and no more than 60 characters.
<b>HTTP Parameter:</b> billingAddress.state  <b>XML Element:</b> <payment> <creditCard> <billingAddress> <state>	<b>State</b>  <b>Presence:</b> Required for Credit Card. <b>Format:</b> Two letters representing the initials of the Brazilian state corresponding (eg, SP).

PARAMETER	DESCRIPTION
<b>HTTP Parameter:</b> billingAddress.country  <b>XML Element:</b> <payment> <creditCard> <billingAddress> <country>	<b>Country</b>  <b>Presence:</b> Required for Credit Card. <b>Format:</b> Recognizes only the BRA value.
<b>SPLIT PARAMETERS</b>	
<b>HTTP Parameter:</b> primaryReceiver.publicKey  <b>XML Element:</b> <checkout> <primaryReceiver> <publicKey>	<b>Primary receiver public key</b>  Specifies the seller that will receive the payment and the transaction will be linked to him. If a primaryReceiver wasn't informed, the application account owner will receive the transaction.  <b>Presence:</b> Optional <b>Type:</b> A valid PagSeguro public key, limited to 40 characters.
<b>HTTP Parameter:</b> receiver[1].publicKey receiver[2].publicKey  <b>XML Element:</b> <checkout> <receivers> <receiver> <publicKey>	<b>Partial receiver public key</b>  Specifies the seller that will receive part of the payment.  <b>Presence:</b> Optional <b>Type:</b> A valid PagSeguro public key, limited to 40 characters.
<b>HTTP Parameter:</b> receiver[1].split.amount receiver[2].split.amount  <b>XML Element:</b> <checkout> <receivers> <receiver> <split> <amount>	<b>Partial receiver amount</b>  Specifies the partial amount destined to this receiver. The sum of this value should be less than or equal to the total amount of the transaction. <b>Presence:</b> Optional <b>Type:</b> Number <b>Format:</b> Numeric with 2 decimal places and separated by a period. Ex.: 9.10

## Return Checkout Transparent API Parameters

CAMPO	DESCRIÇÃO				
<transaction>	This field is the root of the XML and includes the transaction data.				
<transaction> <date>	<p><b>Date of creation of transaction.</b> Specifies the time when the transaction was created.</p> <p><b>Presence:</b> Required <b>Type:</b> Date/time. <b>Format:</b> YYYY-MM-DDThh:mm:ss.sTZD, the official format of W3C for dates. See more about formatting dates.</p>				
<transaction> <lastEventDate>	<p><b>Date of last event.</b> Specifies the time in which the last change in the status of the transaction took place.</p> <p><b>Presence:</b> Required <b>Type:</b> Date/time. <b>Format:</b> YYYY-MM-DDThh:mm:ss.sTZD, the official format of W3C for dates. See more about formatting dates.</p>				
<transaction> <code>	<p><b>ID code of transaction</b> Returns the code that uniquely identifies the transaction.</p> <p><b>Presence:</b> Required <b>Type:</b> Text. <b>Format:</b> A string of 36 characters.</p>				
<transaction> <reference>	<p><b>Reference code of transaction</b> Specifies the code that was used to reference the payment. This code was provided at the time of the payment and is useful to link the PagSeguro transactions to the sales recorded in its system.</p> <p><b>Presence:</b> Optional <b>Type:</b> Text. <b>Format:</b> Free-form, restricted to 200 characters.</p>				
<transaction> <type>	<p><b>Transaction type.</b> Represents the type of transaction received. The more usual values for this field and their respective results are described below.</p> <table> <tr> <th>Code</th><th>Meaning</th></tr> <tr> <td>1</td><td><b>Payment:</b> the transaction was created by a buyer making a payment. This is the most usual type of transaction that you will receive.</td></tr> </table>	Code	Meaning	1	<b>Payment:</b> the transaction was created by a buyer making a payment. This is the most usual type of transaction that you will receive.
Code	Meaning				
1	<b>Payment:</b> the transaction was created by a buyer making a payment. This is the most usual type of transaction that you will receive.				

CAMPO	DESCRIÇÃO																				
	<p>Other less usual types of transactions have been omitted. Take note that new types can be added in future versions of the API.</p> <p><b>Presence:</b> Required.  <b>Type:</b> Number.  <b>Format:</b> Whole number.</p>																				
<transaction> <status>	<p><b>Status of transaction.</b>  Specifies the code representing the transaction status, which allows you to decide if you want to clear the products or services acquired, or not. The possible values are described in the transaction status diagram and are presented together with their respective codes in the table below.</p> <table> <tr> <th>Code</th><th>Meaning</th></tr> <tr> <td>1</td><td><b>Awaiting payment:</b> the buyer initiated the transaction but at this time PagSeguro has not received the payment information.</td></tr> <tr> <td>2</td><td><b>Under analysis:</b> the buyer chose to pay with a credit card and PagSeguro is doing risk analysis on the transaction.</td></tr> <tr> <td>3</td><td><b>Paid:</b> the transaction was paid by the buyer and PagSeguro has received confirmation from the financial institution in charge of the processing.</td></tr> <tr> <td>4</td><td><b>Available:</b> the transaction was paid and has reached the end of the clearing period without having been returned and with no open claims.</td></tr> <tr> <td>5</td><td><b>Under claim:</b> the buyer, within the clearing period of the transaction, opened a claim.</td></tr> <tr> <td>6</td><td><b>Refunded:</b> the transaction value was returned to the buyer.</td></tr> <tr> <td>7</td><td><b>Canceled:</b> the transaction was canceled without having been concluded.</td></tr> <tr> <td>8</td><td><b>Debited:</b> the transaction value was returned to the buyer.</td></tr> <tr> <td>9</td><td><b>Temporary retention:</b> the buyer was opened a request for chargeback with the credit card company.</td></tr> </table>	Code	Meaning	1	<b>Awaiting payment:</b> the buyer initiated the transaction but at this time PagSeguro has not received the payment information.	2	<b>Under analysis:</b> the buyer chose to pay with a credit card and PagSeguro is doing risk analysis on the transaction.	3	<b>Paid:</b> the transaction was paid by the buyer and PagSeguro has received confirmation from the financial institution in charge of the processing.	4	<b>Available:</b> the transaction was paid and has reached the end of the clearing period without having been returned and with no open claims.	5	<b>Under claim:</b> the buyer, within the clearing period of the transaction, opened a claim.	6	<b>Refunded:</b> the transaction value was returned to the buyer.	7	<b>Canceled:</b> the transaction was canceled without having been concluded.	8	<b>Debited:</b> the transaction value was returned to the buyer.	9	<b>Temporary retention:</b> the buyer was opened a request for chargeback with the credit card company.
Code	Meaning																				
1	<b>Awaiting payment:</b> the buyer initiated the transaction but at this time PagSeguro has not received the payment information.																				
2	<b>Under analysis:</b> the buyer chose to pay with a credit card and PagSeguro is doing risk analysis on the transaction.																				
3	<b>Paid:</b> the transaction was paid by the buyer and PagSeguro has received confirmation from the financial institution in charge of the processing.																				
4	<b>Available:</b> the transaction was paid and has reached the end of the clearing period without having been returned and with no open claims.																				
5	<b>Under claim:</b> the buyer, within the clearing period of the transaction, opened a claim.																				
6	<b>Refunded:</b> the transaction value was returned to the buyer.																				
7	<b>Canceled:</b> the transaction was canceled without having been concluded.																				
8	<b>Debited:</b> the transaction value was returned to the buyer.																				
9	<b>Temporary retention:</b> the buyer was opened a request for chargeback with the credit card company.																				

CAMPO	DESCRIÇÃO										
	<p>Other less relevant statuses have been omitted. In short, you must consider transactions in the <b>Paid</b> status for the clearing of products or services.</p> <p><b>Presence:</b> Required.  <b>Type:</b> Number.  <b>Format:</b> Whole number.</p>										
<transaction> <cancellationSource>	<p><b>Origin of cancellation.</b>  Specifies the origin of the cancellation of the transaction: by financial institutions (Issuing Bank or Card Operator) or by PagSeguro.</p> <table border="1"> <thead> <tr> <th>Code</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>INTERNAL</td><td><b>PagSeguro</b></td></tr> <tr> <td>EXTERNAL</td><td><b>Financial Institutions</b></td></tr> </tbody> </table> <p><b>Presence:</b> Optional (only when transactionStatus is equal to 7).  <b>Type:</b> Text.  <b>Format:</b> Possible INTERNAL or EXTERNAL values.</p>	Code	Meaning	INTERNAL	<b>PagSeguro</b>	EXTERNAL	<b>Financial Institutions</b>				
Code	Meaning										
INTERNAL	<b>PagSeguro</b>										
EXTERNAL	<b>Financial Institutions</b>										
<transaction> <paymentMethod>	<p>Data of payment method used by buyer.</p>										
<transaction> <paymentMethod> <type>	<p><b>Type of payment method.</b>  Specifies the type of payment method used by the buyer. This type groups several payment methods and generally defines the behavior of the transaction. The table below describes the values available and their meanings.</p> <table border="1"> <thead> <tr> <th>Code</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>1</td><td><b>Credit card:</b> the buyer chose to pay the transaction using a credit card.</td></tr> <tr> <td>2</td><td><b>“Boleto” payment slip:</b> the buyer chose to pay with a bank payment slip (“boleto”).</td></tr> <tr> <td>3</td><td><b>Online debit (TEF):</b> the buyer chose to pay the transaction with online debit of some partner banks.</td></tr> <tr> <td>4</td><td><b>PagSeguro Balance:</b> the buyer chose to pay for the transaction using the balance of his/her PagSeguro account.</td></tr> </tbody> </table>	Code	Meaning	1	<b>Credit card:</b> the buyer chose to pay the transaction using a credit card.	2	<b>“Boleto” payment slip:</b> the buyer chose to pay with a bank payment slip (“boleto”).	3	<b>Online debit (TEF):</b> the buyer chose to pay the transaction with online debit of some partner banks.	4	<b>PagSeguro Balance:</b> the buyer chose to pay for the transaction using the balance of his/her PagSeguro account.
Code	Meaning										
1	<b>Credit card:</b> the buyer chose to pay the transaction using a credit card.										
2	<b>“Boleto” payment slip:</b> the buyer chose to pay with a bank payment slip (“boleto”).										
3	<b>Online debit (TEF):</b> the buyer chose to pay the transaction with online debit of some partner banks.										
4	<b>PagSeguro Balance:</b> the buyer chose to pay for the transaction using the balance of his/her PagSeguro account.										

CAMPO	DESCRIÇÃO																
	<table> <tr> <td>5</td><td><b>Oi Paggo*</b>: the buyer chose to pay the transaction through his/her Oi mobile phone.</td></tr> <tr> <td>7</td><td><b>Account deposit</b>: the buyer chose to make a deposit in a PagSeguro bank account. He/she will have to go to a bank, make the deposit, keep the payment slip and return it to PagSeguro. The transaction will be confirmed only after this process has been concluded, which can take 2 to 13 business days.</td></tr> </table> <p>* The marked types are not available for use.</p> <p><b>Presence:</b> Required.  <b>Type:</b> Number.  <b>Format:</b> Whole number.</p>	5	<b>Oi Paggo*</b> : the buyer chose to pay the transaction through his/her Oi mobile phone.	7	<b>Account deposit</b> : the buyer chose to make a deposit in a PagSeguro bank account. He/she will have to go to a bank, make the deposit, keep the payment slip and return it to PagSeguro. The transaction will be confirmed only after this process has been concluded, which can take 2 to 13 business days.												
5	<b>Oi Paggo*</b> : the buyer chose to pay the transaction through his/her Oi mobile phone.																
7	<b>Account deposit</b> : the buyer chose to make a deposit in a PagSeguro bank account. He/she will have to go to a bank, make the deposit, keep the payment slip and return it to PagSeguro. The transaction will be confirmed only after this process has been concluded, which can take 2 to 13 business days.																
<transaction> <paymentLink>	<p><b>Payment Link.</b>  Specifies the URL to view the “boleto” payment slip or, when the payment method is TEF, the URL to open the bank pop-up window. When the payment method is Credit Card this parameter will be omitted.</p> <p><b>Presence:</b> Only for payments via “Boleto” and TEF.  <b>Type:</b> Text.  <b>Format:</b> URL</p>																
<transaction> <paymentMethod> <code>	<p><b>ID code of payment method</b>  Specifies the code that identifies the payment method used by the buyer. The payment method describes the brand of the credit card used or the bank chosen for online debit. The table below describes the possible values and their meanings.</p> <table> <tr> <th>Code</th><th>Meaning</th></tr> <tr> <td>101</td><td>Visa credit card.</td></tr> <tr> <td>102</td><td>Mastercard credit card.</td></tr> <tr> <td>103</td><td>American Express credit card.</td></tr> <tr> <td>104</td><td>Diners credit card.</td></tr> <tr> <td>105</td><td>Hipercard credit card.</td></tr> <tr> <td>106</td><td>Aura credit card.</td></tr> <tr> <td>107</td><td>Elo credit card.</td></tr> </table>	Code	Meaning	101	Visa credit card.	102	Mastercard credit card.	103	American Express credit card.	104	Diners credit card.	105	Hipercard credit card.	106	Aura credit card.	107	Elo credit card.
Code	Meaning																
101	Visa credit card.																
102	Mastercard credit card.																
103	American Express credit card.																
104	Diners credit card.																
105	Hipercard credit card.																
106	Aura credit card.																
107	Elo credit card.																



CAMPO	DESCRIÇÃO	
	108	PLENOcard credit card.
	109	PersonalCard credit card.
	110	JCB credit card.
	111	Discover credit card.
	112	BrasilCard credit card.
	113	FORTBRASIL credit card.
	114	CARDBAN credit card.
	115	VALECARD credit card.
	116	Cabal credit card.
	117	Mais! credit card.
	118	Avista credit card.
	119	GRANDCARD credit card.
	201	Bradesco “boleto”*
	202	Santander “boleto”.
	301	Bradesco online debit.
	302	Itau online debit.
	303	Unibanco online debit.*
	304	Banco do Brasil online debit.
	305	Banco Real online debit.*
	306	Banrisul online debit.
	307	HSBC online debit.
	401	PagSeguro balance.
	501	Oi Paggo.*
	701	Deposit in account – Banco do Brasil
	* The marked payment methods are not available for use.	

CAMPO	DESCRIÇÃO
	<p><b>Presence:</b> Required.</p> <p><b>Type:</b> Number.</p> <p><b>Format:</b> Whole number.</p>
<transaction> <grossAmount>	<p><b>Gross value of transaction.</b>  Specifies the gross value of the transaction, calculated by adding the prices of all the items in the payment.</p> <p><b>Presence:</b> Required.</p> <p><b>Type:</b> Number.</p> <p><b>Format:</b> Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.</p>
<transaction> <discountAmount>	<p><b>Value of discount granted.</b>  Specifies the discount value granted to the buyers who choose to pay with online debit or “boleto.” This discount applies when you choose to include in the price of the products the cost of the payment installments using credit card. The discount is granted in order not to charge buyers who chose to pay up front.</p> <p><b>Presence:</b> Required.</p> <p><b>Type:</b> Number.</p> <p><b>Format:</b> Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.</p>
<transaction> <creditorFees>	Fees charged data
<transaction> <creditorFees> <installmentFeeAmount>	<p><b>Installment amount fees.</b>  Specifies the total value of the installment fees charged by PagSeguro in this transaction.</p> <p><b>Presence:</b> Required.</p> <p><b>Type:</b> Number.</p> <p><b>Format:</b> Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.</p>
<transaction> <creditorFees> <operationalFeeAmount>	<p><b>Operational amount fees.</b>  Specifies the total value of the operational fees charged by PagSeguro in this transaction.</p> <p><b>Presence:</b> Required.</p> <p><b>Type:</b> Number.</p> <p><b>Format:</b> Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.</p>

CAMPO	DESCRIÇÃO
<transaction> <netAmount>	<p><b>Net value of transaction.</b> Specifies the net value of the transaction that represents the gross value, minus the fee value. At this time, the value of extra amount (which can be positive or negative) is also considered in the calculation.</p> <p><b>Presence:</b> Required.  <b>Type:</b> Number.  <b>Format:</b> Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.</p>
<transaction> <escrowEndDate>	<p><b>Credit date.</b> Date in which the transaction value will be available in the seller’s account.</p> <p><b>Presence:</b> Present only when the transaction status is one of the following values: Paid (3), Available (4), Under claim (5) or Returned (6).  <b>Type:</b> Date/Time.  <b>Format:</b> YYYY-MM-DDThh:mm:ss.sTZD, the official format of W3C for dates. See more about formatting dates.</p>
<transaction> <extraAmount>	<p><b>Extra amount.</b> Specifies the extra amount that was added or subtracted from the amount paid by the buyer. This value is specified by you in the payment and can represent a value that you want to charge additionally to the buyer or a discount that you want to grant to the buyer.</p> <p><b>Presence:</b> Required.  <b>Type:</b> Number.  <b>Format:</b> Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.</p>
<transaction> <installmentCount>	<p><b>Amount of installments.</b> Indicates the number of installments that the buyer chose in the payment using a credit card.</p> <p><b>Presence:</b> Required.  <b>Type:</b> Number.  <b>Format:</b> Whole number.</p>
<transaction> <itemCount>	<p><b>Number of items in transaction.</b> Indicates out the number of items contained in this transaction.</p> <p><b>Presence:</b> Required.  <b>Type:</b> Number.  <b>Format:</b> Whole number.</p>
<transaction> <items>	<p>List of items contained in the transaction. The number of items under this item represent the value of the item count.</p>

CAMPO	DESCRIÇÃO
<transaction> <items> <item>	Represents an item of the transaction.
<transaction> <items> <item> <id>	<b>ID of item.</b> Identifies the item of the transaction. This identifier must be unique per transaction and must be informed in the payment flow.  <b>Presence:</b> Required. <b>Type:</b> Text. <b>Format:</b> Free-form.
<transaction> <items> <item> <description>	<b>Item description.</b> Describes a single item in the transaction. The same description that was provided during the payment flow.  <b>Presence:</b> Required. <b>Type:</b> Text. <b>Format:</b> Free-form.
<transaction> <items> <item> <amount>	<b>Unit value of item.</b> Specifies the unit price of the item of the transaction. This is the value that was specified in the payment flow.  <b>Presence:</b> Required. <b>Type:</b> Number. <b>Format:</b> Decimal, with two decimal places separated by a period ("."). For example, 1234.56.
<transaction> <items> <item> <quantity>	<b>Quantity of item.</b> Specifies the quantity of the item in the transaction. This is the quantity that was specified in the payment flow.  <b>Presence:</b> Required. <b>Type:</b> Number. <b>Format:</b> A whole number greater or equal to 1 and less or equal to 999.
<transaction> <sender>	<b>Buyer data.</b>
<transaction> <sender> <email>	<b>Buyer's email.</b> Specifies the email of the buyer who made the transaction  <b>Presence:</b> Required. <b>Type:</b> Text. <b>Format:</b> A valid email (ex., ), restricted to 60 characters.
<transaction> <sender>	<b>Buyer's full name.</b> Specifies the full name of the buyer who made the payment.

CAMPO	DESCRIÇÃO								
<name>	<b>Presence:</b> Optional <b>Type:</b> Text <b>Format:</b> At least two strings of characters, restricted to a total of 50 characters.								
<transaction> <sender> <phone>	<b>Buyer telephone data</b>								
<transaction> <sender> <phone> <areaCode>	<b>Area code (DDD) of buyer.</b> Specifies the area code (DDD) of buyer who made the payment  <b>Presence:</b> Optional <b>Type:</b> Number <b>Format:</b> A 2-digit number that represents a valid area code.								
<transaction> <sender> <phone> <number>	<b>Telephone number of buyer.</b> Specifies the telephone number of buyer who made the payment  <b>Presence:</b> Optional <b>Type:</b> Number <b>Format:</b> A number comprised of 7 to 9 digits.								
<transaction> <shipping>	<b>Shipping data.</b>								
<transaction> <shipping> <type>	<b>Shipping type.</b>  Specifies the type of shipping that will be used to send the product. The table below Specifies the possible values and their meanings. <table border="1"> <thead> <tr> <th>Code</th><th>Meaning</th></tr> </thead> <tbody> <tr> <td>1</td><td>Normal delivery (PAC)</td></tr> <tr> <td>2</td><td>SEDEX special delivery.</td></tr> <tr> <td>3</td><td>Unspecified shipping type.</td></tr> </tbody> </table> <b>Presence:</b> Required. <b>Type:</b> Number <b>Format:</b> Whole.	Code	Meaning	1	Normal delivery (PAC)	2	SEDEX special delivery.	3	Unspecified shipping type.
Code	Meaning								
1	Normal delivery (PAC)								
2	SEDEX special delivery.								
3	Unspecified shipping type.								
<transaction>	<b>Total shipping cost.</b>								

CAMPO	DESCRIÇÃO
<shipping> <cost>	<p>Specifies the total shipping cost, based on the shipping options informed in the payment flow.</p> <p><b>Presence:</b> Required.  <b>Type:</b> Number  <b>Format:</b> Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.</p>
<transaction> <shipping> <address>	<p><b>Shipping address data.</b></p>
<transaction> <shipping> <address> <country>	<p><b>Country of shipping address</b>            Specifies the country of the shipping address of the product.</p> <p><b>Presence:</b> Optional  <b>Type:</b> Text  <b>Format:</b> At this time, only the <b>BRA</b> value is allowed.</p>
<transaction> <shipping> <address> <state>	<p><b>State of shipping address</b>            Specifies the state of the shipping address of the product.</p> <p><b>Presence:</b> Optional  <b>Type:</b> Text  <b>Format:</b> Two letters, representing the abbreviation of the corresponding Brazilian state.</p>
<transaction> <shipping> <address> <city>	<p><b>City of shipping address</b>            Specifies the city of the shipping address of the product.</p> <p><b>Presence:</b> Optional  <b>Type:</b> Text  <b>Format:</b> Free-form. Must be a valid Brazilian city name, in accordance with Post Office data.</p>
<transaction> <shipping> <address> <postalCode>	<p><b>Postal code of shipping address</b>            Specifies the postal code of the shipping address of the product.</p> <p><b>Presence:</b> Optional  <b>Type:</b> Number  <b>Format:</b> An 8-digit number.</p>
<transaction> <shipping> <address> <district>	<p><b>District of shipping address</b>            Specifies the district of the shipping address of the product.</p> <p><b>Presence:</b> Optional  <b>Type:</b> Text  <b>Format:</b> Free-form.</p>

CAMPO	DESCRIÇÃO
<transaction> <shipping> <address> <street>	<b>Street name of shipping address</b> Specifies the street name of the shipping address of the product.  <b>Presence:</b> Optional <b>Type:</b> Text <b>Format:</b> Free-form.
<transaction> <shipping> <address> <number>	<b>Number of shipping address.</b> Specifies the number of the shipping address of the product.  <b>Presence:</b> Optional <b>Type:</b> Text <b>Format:</b> Free-form.
<transaction> <shipping> <address> <complement>	<b>Complement of shipping address.</b> Specifies the complement of the shipping address of the product.  <b>Presence:</b> Optional <b>Type:</b> Text <b>Format:</b> Free-form.

## Error list

In case of your application send an invalid or out of pattern data, a message will be returned with a message explaining the problem. Check below the errors that can be returned:

### HTTP 401 - Unauthorized

Occurs when your application send an invalid or inexistent credential (appld, appkey or authorizationCode).

### HTTP 405 – Method Not Allowed

Occurs when your application using an unexpected HTTP method. Check if the expected method is a GET or POST.

### HTTP 415 – Cannot consume content type

Check if the Content-Type of the API call is correct.

### HTTP 400 – Bad Request

Occurs when a problem in one or more XML parameters is detected. This return has an XML in your body that identify the errors found. Check the list of errors below:

CÓDGO	DESCRIÇÃO
10000	invalid creditcard brand.
10001	creditcard number with invalid length.
10002	invalid date format.

CÓDIGO	DESCRIÇÃO
10003	invalid security field.
10004	cvv is mandatory.
10006	security field with invalid length.
53004	items invalid quantity.
53005	currency is required.
53006	currency invalid value: {0}
53007	reference invalid length: {0}
53008	notificationURL invalid length: {0}
53009	notificationURL invalid value: {0}
53010	sender email is required.
53011	sender email invalid length: {0}
53012	sender email invalid value: {0}
53013	sender name is required.
53014	sender name invalid length: {0}
53015	sender name invalid value: {0}
53017	sender cpf invalid value: {0}
53018	sender area code is required.
53019	sender area code invalid value: {0}
53020	sender phone is required.
53021	sender phone invalid value: {0}
53022	shipping address postal code is required.
53023	shipping address postal code invalid value: {0}
53024	shipping address street is required.
53025	shipping address street invalid length: {0}
53026	shipping address number is required.



CÓDIGO	DESCRIÇÃO
53027	shipping address number invalid length: {0}
53028	shipping address complement invalid length: {0}
53029	shipping address district is required.
53030	shipping address district invalid length: {0}
53031	shipping address city is required.
53032	shipping address city invalid length: {0}
53033	shipping address state is required.
53034	shipping address state invalid value: {0}
53035	shipping address country is required.
53036	shipping address country invalid length: {0}
53037	credit card token is required.
53038	installment quantity is required.
53039	installment quantity invalid value: {0}
53040	installment value is required.
53041	installment value invalid value: {0}
53042	credit card holder name is required.
53043	credit card holder name invalid length: {0}
53044	credit card holder name invalid value: {0}
53045	credit card holder cpf is required.
53046	credit card holder cpf invalid value: {0}
53047	credit card holder birthdate is required.
53048	credit card holder birthdate invalid value: {0}
53049	credit card holder area code is required.
53050	credit card holder area code invalid value: {0}
53051	credit card holder phone is required.

CÓDIGO	DESCRIÇÃO
53052	credit card holder phone invalid value: {0}
53053	billing address postal code is required.
53054	billing address postal code invalid value: {0}
53055	billing address street is required.
53056	billing address street invalid length: {0}
53057	billing address number is required.
53058	billing address number invalid length: {0}
53059	billing address complement invalid length: {0}
53060	billing address district is required.
53061	billing address district invalid length: {0}
53062	billing address city is required.
53063	billing address city invalid length: {0}
53064	billing address state is required.
53065	billing address state invalid value: {0}
53066	billing address country is required.
53067	billing address country invalid length: {0}
53068	receiver email invalid length: {0}
53069	receiver email invalid value: {0}
53070	item id is required.
53071	item id invalid length: {0}
53072	item description is required.
53073	item description invalid length: {0}
53074	item quantity is required.
53075	item quantity out of range: {0}
53076	item quantity invalid value: {0}

CÓDIGO	DESCRIÇÃO
53077	item amount is required.
53078	item amount invalid pattern: {0}. Must fit the pattern: \\d+\\.\\d{2\\}
53079	item amount out of range: {0}
53081	sender is related to receiver.
53084	invalid receiver: {0}, verify receiver's account status and if it is a seller's account.
53085	payment method unavailable.
53086	cart total amount out of range: {0}
53087	invalid credit card data.
53091	sender hash invalid.
53092	credit card brand is not accepted.
53095	shipping type invalid pattern: {0}
53096	shipping cost invalid pattern: {0}
53097	shipping cost out of range: {0}
53098	cart total value is negative: {0}
53099	extra amount invalid pattern: {0}. Must fit the pattern: -?\\d+\\.\\d{2\\}
53101	payment mode invalid value, valid values are default and gateway.
53102	payment method invalid value, valid values are creditCard, boleto e eft.
53104	shipping cost was provided, shipping address must be complete.
53105	sender information was provided, email must be provided too.
53106	credit card holder is incomplete.
53109	shipping address information was provided, sender email must be provided too.
53110	eft bank is required.
53111	eft bank is not accepted.
53115	sender born date invalid value: {0}
53117	sender cnpj invalid value: {0}

CÓDIGO	DESCRIÇÃO
53122	sender email invalid domain: {0}. You must use an email @sandbox.pagseguro.com.br
53140	installment quantity out of range: {0}. The value must be greater than zero.
53141	sender is blocked.
53142	credit card token invalid.

## Attachments

### API call example for Boleto transaction

```

1. <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2. <payment>
3.   <mode>default</mode>
4.   <method>boleto</method>
5.   <sender>
6.     <name>Customer Name</name>
7.     <email>customer@uol.com.br</email>
8.     <phone>
9.       <areaCode>11</areaCode>
10.      <number>30380000</number>
11.    </phone>
12.    <documents>
13.      <document>
14.        <type>CPF</type>
15.        <value>11475714734</value>
16.      </document>
17.    </documents>
18.    <hash>c162e6131dbb9696e8b49147a9c5eacabac0cdca91458ee5177819d0a3c9d565</hash>
19.  </sender>
20.  <currency>BRL</currency>
21.  <notificationURL>https://yourstore.com.br/notification</notificationURL>
22.  <items>
23.    <item>
24.      <id>1</id>
25.      <description>Product 1 Description</description>
26.      <quantity>2</quantity>
27.      <amount>100.00</amount>
28.    </item>
29.  </items>
30.  <extraAmount>0.00</extraAmount>
31.  <reference>ORDER123</reference>
32.  <shipping>
33.    <address>
34.      <street>Av. Brigadeiro Faria Lima</street>
35.      <number>1384</number>
36.      <complement>1 andar</complement>
37.      <district>Jardim Paulistano</district>
38.      <city>Sao Paulo</city>

```

```

39.         <state>SP</state>
40.         <country>BRA</country>
41.         <postalCode>01452002</postalCode>
42.     </address>
43.     <type>3</type>
44.     <cost>0.00</cost>
45. </shipping>
46. <primaryReceiver>
47.     <publicKey>PUBCF30546BDF334CC3A59D772E3636D8D3</publicKey>
48. </primaryReceiver>
49. <receivers>
50.     <receiver>
51.         <publicKey>PUBCF30546BDF334CC3A59D772E3636D8D4</publicKey>
52.         <split>
53.             <amount>20.00</amount>
54.             <ratePercent>50.00</ratePercent>
55.             <feePercent>50.00</feePercent>
56.         </split>
57.     </receiver>
58.     <receiver>
59.         <publicKey>PUBCF30546BDF334CC3A59D772E3636D8D5</publicKey>
60.         <split>
61.             <amount>30.00</amount>
62.             <ratePercent>50.00</ratePercent>
63.             <feePercent>50.00</feePercent>
64.         </split>
65.     </receiver>
66. </receivers>
67. </payment>

```

## API call example for Online Debit

```

1. <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2. <payment>
3.     <mode>default</mode>
4.     <method>eft</method>
5.     <bank>
6.         <name>itau</name>
7.     </bank>
8.     <sender>
9.         <name>Customer Name</name>
10.        <email>customer@uol.com.br</email>
11.        <phone>
12.            <areaCode>11</areaCode>
13.            <number>30380000</number>
14.        </phone>
15.        <documents>
16.            <document>
17.                <type>CPF</type>
18.                <value>11475714734</value>
19.            </document>
20.        </documents>
21.        <hash>c162e6131dbb9696e8b49147a9c5eacabac0cdca91458ee5177819d0a3c9d565</hash>
22.    </sender>
23.    <currency>BRL</currency>

```

```

24. <notificationURL>https://yourstore.com.br/notification</notificationURL>
25. <items>
26.   <item>
27.     <id>1</id>
28.     <description>Product 1 Description</description>
29.     <quantity>2</quantity>
30.     <amount>100.00</amount>
31.   </item>
32. </items>
33. <extraAmount>0.00</extraAmount>
34. <reference>ORDER123</reference>
35. <shipping>
36.   <address>
37.     <street>Av. Brigadeiro Faria Lima</street>
38.     <number>1384</number>
39.     <complement>1 andar</complement>
40.     <district>Jardim Paulistano</district>
41.     <city>Sao Paulo</city>
42.     <state>SP</state>
43.     <country>BRA</country>
44.     <postalCode>01452002</postalCode>
45.   </address>
46.   <type>3</type>
47.   <cost>0.00</cost>
48. </shipping>
49. <primaryReceiver>
50.   <publicKey>PUBCF30546BDF334CC3A59D772E3636D8D3</publicKey>
51. </primaryReceiver>
52. <receivers>
53.   <receiver>
54.     <publicKey>PUBCF30546BDF334CC3A59D772E3636D8D4</publicKey>
55.     <split>
56.       <amount>20.00</amount>
57.     </split>
58.   </receiver>
59.   <receiver>
60.     <publicKey>PUBCF30546BDF334CC3A59D772E3636D8D5</publicKey>
61.     <split>
62.       <amount>30.00</amount>
63.       <ratePercent>50.00</ratePercent>
64.       <feePercent>50.00</feePercent>
65.     </split>
66.   </receiver>
67. </receivers>
68. </payment>

```

## API call example for Credit Card

```

1. <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2. <payment>
3.   <mode>default</mode>
4.   <method>creditCard</method>
5.   <sender>
6.     <name>Customer Name</name>
7.     <email>customer@uol.com.br</email>
8.     <phone>
9.       <areaCode>11</areaCode>
10.      <number>30380000</number>
11.    </phone>
12.    <documents>
13.      <document>
14.        <type>CPF</type>
15.        <value>11475714734</value>
16.      </document>
17.    </documents>
18.    <hash>c162e6131dbb9696e8b49147a9c5eacabac0cdca91458ee5177819d0a3c9d565</hash>
19.  </sender>
20.  <currency>BRL</currency>
21.  <notificationURL>https://yourstore.com.br/notification</notificationURL>
22.  <items>
23.    <item>
24.      <id>1</id>
25.      <description>Product 1 Description</description>
26.      <quantity>2</quantity>
27.      <amount>100.00</amount>
28.    </item>
29.  </items>
30.  <extraAmount>0.00</extraAmount>
31.  <reference>ORDER123</reference>
32.  <shipping>
33.    <address>
34.      <street>Av. Brigadeiro Faria Lima</street>
35.      <number>1384</number>
36.      <complement>1 andar</complement>
37.      <district>Jardim Paulistano</district>
38.      <city>Sao Paulo</city>
39.      <state>SP</state>
40.      <country>BRA</country>
41.      <postalCode>01452002</postalCode>
42.    </address>
43.    <type>3</type>
44.    <cost>0.00</cost>
45.  </shipping>
46.  <creditCard>
47.    <token>4a56sd456a4d54asd65as4d56a4sd564</token>
48.    <installment>
49.      <quantity>2</quantity>
50.      <value>100.00</value>
51.    </installment>
52.    <holder>
53.      <name>Credit Card Holder Name</name>

```

```
54.         <documents>
55.             <document>
56.                 <type>CPF</type>
57.                 <value>00722333665</value>
58.             </document>
59.         </documents>
60.         <birthDate>20/10/1980</birthDate>
61.         <phone>
62.             <areaCode>11</areaCode>
63.             <number>999991111</number>
64.         </phone>
65.     </holder>
66.     <billingAddress>
67.         <street>Av. Brigadeiro Faria Lima</street>
68.         <number>1384</number>
69.         <complement>1 andar</complement>
70.         <district>Jardim Paulistano</district>
71.         <city>Sao Paulo</city>
72.         <state>SP</state>
73.         <country>BRA</country>
74.         <postalCode>01452002</postalCode>
75.     </billingAddress>
76. </creditCard>
77. <primaryReceiver>
78.     <publicKey>PUBCF30546BDF334CC3A59D772E3636D8D3</publicKey>
79. </primaryReceiver>
80. <receivers>
81.     <receiver>
82.         <publicKey>PUBCF30546BDF334CC3A59D772E3636D8D4</publicKey>
83.         <split>
84.             <amount>20.00</amount>
85.         </split>
86.     </receiver>
87.     <receiver>
88.         <publicKey>PUBCF30546BDF334CC3A59D772E3636D8D5</publicKey>
89.         <split>
90.             <amount>30.00</amount>
91.         </split>
92.     </receiver>
93. </receivers>
94. </payment>
```