Transparent Checkout Integration Guide

Revision History

DATE	DESCRIPTION
05/Jun/2014	Overview

Copyright

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

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).

Transparent Checkout is available for Seller and Enterprise account types. The following sections will show how to integrate your payment system with PagSeguro Transparent Checkout.

Authentication in PagSeguro APIs

All calls to the APIs on the domain ws.pagseguro.uol.com.br require authentication.
Authentication can be done by Seller or by application.
The following HTTP parameters must be sent in all service calls to the aforementioned domain.

Authentication for Sellers

Use this type of authentication if you are creating sales for your own store.

HTTP PARAMETER	DESCRIPTION
email	Email of seller **Presence:** Required. **Type:** Text **Format:** A valid e-mail, restricted to 60 characters. The e-mail provided must be linked to the PagSeguro account that is making the API calls
token	Token of seller **Presence:** Required. **Type:** Text **Example:** 861D0B48DBA3485AB2E36675332F2812

Autenticação Para Aplicações

Se você é uma plataforma ou uma aplicação e está integrando várias lojas com o Modelo de Aplicações do PagSeguro, então deve utilizar esse tipo de autenticação.

PARÂMETRO HTTP	DESCRIÇÃO
appId	ID of Application **Presence:** Required. **Type**: Text **Format:** A string of no more than 60 characters. **Example:** pagseguro_uol
Appkey	Specifies the token that corresponds to the PagSeguro application that is making the request. **Presence:** Required. **Type:** Text **Format:** A string of 32 characters. **Example:** 861D0B48DBA3485AB2E36675332F2812
authorizationCode	Code that authorizes the application to take action in the name of the seller. **Presence:** Required. **Type:** Text **Example:** 5A220D39B3B31E8BB4E3EFA4FCC4E6L5

Transparent Checkout

The API for Transparent Checkout offers more control and flexibility over the payment process. With this integration, the customer stays within an e-commerce website environment throughout the whole purchase process, without needing to visit intermediate payment pages. For this integration, PagSeguro created a few services that together create an integrated and safe checkout process that is invisible to the customer.

Integration

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

- Start a payment session
- 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.

This description is valid for calls that use **HTTP** parameters as well as for a calls using the **XML** format.
This API requires the authentication described on **pages 4-5**.

URL:
HttpMethod: POST
`https://ws.pagseguro.uol.com.br/v2/sessions`

Example of Seller authentication:
`https://ws.pagseguro.uol.com.br/v2/sessions?email={email}&token={token}`

Example of Application authentication:
`https://ws.pagseguro.uol.com.br/v2/sessions?authorizationCode={authorizationCode}&appId={appId}&appKey={appKey}`

Return:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<session>
  <id>620f99e348c24f07877c927b353e49d3</id>
</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 be 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>
<script type="text/javascript">
  PagSeguroDirectPayment.setSessionId('ID DE SESSÃO GERADO NA PÁGINA 6);
</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 as described on **page 6.**

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 ID

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();
```

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

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 returns various pieces of information: the brand, the size of the CVV security code, if expiration date is applicable, and the which validation algorithm applies. Calling this service is not Required.

Syntax:

```
PagSeguroDirectPayment.getBrand ({
cardBin: {BIN of card number},
success: {function of callback for successful calls},
error: {callback function for calls that failed},
complete: {callback function for all calls}
});
```

Example:

```
PagSeguroDirectPayment.getBrand ({
cardBin: $(“input#cartao”), val ( ),
success: function(response) {
// banner found
},
Error: function (response) {
//error handling
},
Complete: function(response) {
//same handling for all calls
}
});
```

Return:

```
{
   "brand":{
      "name":"visa",
      "bin":411111,
      "cvvSize":3,
      "expirable":true,
      "validationAlgorithm":"LUHN"
   }
}
```

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 (**required**).

Syntax:

```
PagSeguroDirectPayment.createCardToken({
  cardNumber: { number},
  brand: { brand },
  cvv: { security code },
  expirationMonth: { month of expiration },
  expirationYear: { year of expiration },
  success: { callback function for successful calls}
  error: { callback function for calls that failed },
  complete: { callback function for all calls }
});
```

Exemplo:

```
var param = {
  cardNumber: $("input#cartao").val(),
  cvv: $("input#cvv").val(),
  expirationMonth: $("input#validadeMes").val(),
  expirationYear: $("input#validadeAno").val(),
  success: function(response) {
     // token generated, this must be used in the API call to the Transparent Checkout
},
  error: function(response) {
      //error treatment
    },
  complete: function(response) {
      // same handling for all calls
    }
}

// optional parameter for any call
if($("input#bandeira").val() != '') {
  param.brand = $("input#bandeira").val();
}

PagSeguroDirectPayment.createCardToken(param);
```

Return:

```
{
   "card":{
      "token":"653fe9044cf149f9b7db562431cb130d"
   }
}
```

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**) 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).

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:

```
PagSeguroDirectPayment.getInstallments({
  amount: { payment amount },
  brand: { card banner },
  success: { callback function for successful calls },
  error: { callback function for calls that failed },
  complete: { callback function for all calls }
});
```

Example:

```
PagSeguroDirectPayment.getInstallments({
  amount: $("input#valorPagto").val(),
  brand: $("input#bandeira").val(),
  success: function(response) {
      // available installment options
    },
  error: function(response) {
      // error handling
    },
  complete: function(response) {
      // same handling for all calls
    }
});
```

Return:

```
{
   "error":false,
   "installments":{
      "visa":[
         {
            "quantity":1,
            "totalAmount":16,
            "installmentAmount":16,
            "interestFree":true
         },{
            "quantity":2,
            "totalAmount":16.48,
            "installmentAmount":8.24,
            "interestFree":false
         },{
            "quantity":3,
            "totalAmount":16.64,
            "installmentAmount":5.55,
            "interestFree":false
         }
      ]
   }
}
```

Transparent Checkout API

This service sends the buyer's data and the payment information for the transaction.

URL:
HttpMethod: POST
`https://ws.pagseguro.uol.com.br/v2/transactions`

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.

For example: the parameters **itemId1**, **itemDescription1**, **itemAmount1** and **itemQuantity1** refer to the first item being purchased, while parameters **itemId2**, **itemDescription2**, **itemAmount2** and **itemQuantity2** refer to the second item.

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).

```
curl https://ws.pagseguro.uol.com.br/v2/transactions/ -d\
    "email=suporte@lojamodelo.com.br\
    &token=95112EE828D94278BD394E91C4388F20\
    &paymentMode=default\
    &paymentMethod=creditCard\
    &receiverEmail=suporte@lojamodelo.com.br\
    &currency=BRL\
    &extraAmount=1.00
    &itemId1=0001\
    &itemDescription1=Notebook Prata\
    &itemAmount1=24300.00\
    &itemQuantity1=1\
    &itemId2=0002\
    &itemDescription2=Notebook Rosa\
    &itemAmount2=25600.00\
    &itemQuantity2=2\
    &notificationURL=https://sualoja.com.br/notifica.html\
    &reference=REF1234\
    &senderName=Jose Comprador\
    &senderCPF=22111944785\
    &senderAreaCode=11\
    &senderPhone=56273440\
    &senderEmail=comprador@uol.com.br\
    &senderHash=abc123\
    &shippingAddressStreet=Av. Brig. Faria Lima\
    &shippingAddressNumber=1384\
    &shippingAddressComplement=5o andar\
    &shippingAddressDistrict=Jardim Paulistano\
    &shippingAddressPostalCode=01452002\
    &shippingAddressCity=Sao Paulo\
    &shippingAddressState=SP\
    &shippingAddressCountry=BRA
    &shippingType=1
    &shippingCost=1.00
    &creditCardToken=4as56d4a56d456as456dsa\
    &installmentQuantity=5\
    &installmentValue=125.22\
    &creditCardHolderName=Jose Comprador\
    &creditCardHolderCPF=22111944785\
    &creditCardHolderBirthDate=27/10/1987\
    &creditCardHolderAreaCode=11\
    &creditCardHolderPhone=56273440
    &billingAddressStreet=Av. Brig. Faria Lima\
    &billingAddressNumber=1384\
    &billingAddressComplement=5o andar\
    &billingAddressDistrict=Jardim Paulistano\
    &billingAddressPostalCode=01452002\
    &billingAddressCity=Sao Paulo\
    &billingAddressState=SP\
    &billingAddressCountry=BRA\"
```

The example below shows data sent as **XML**.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<payment>
    <mode>default</mode>
    <method>creditCard</method>
    <sender>
        <name>Fulano Silva</name>
        <email>fulano.silva@uol.com.br</email>
        <phone>
            <areaCode>11</areaCode>
            <number>30380000</number>
        </phone>
        <documents>
            <document>
                <type>CPF</type>
                <value>11475714734</value>
            </document>
        </documents>
        <hash>abc1234</hash>
    </sender>
    <currency>BRL</currency>
    <notificationURL>https://sualoja.com.br/notifica.html</notificationURL>
    <items>
        <item>
            <id>1</id>
            <description>Descricao do item a ser vendido</description>
            <quantity>2</quantity>
            <amount>1.00</amount>
        </item>
        <item>
            <id>2</id>
            <description>Descrição do item a ser vendido</description>
            <quantity>2</quantity>
            <amount>1.00</amount>
        </item>
        <item>
            <id>3</id>
            <description>Descrição do item a ser vendido</description>
            <quantity>2</quantity>
            <amount>1.00</amount>
        </item>
    </items>
    <extraAmount>0.00</extraAmount>
    <reference>R123456</reference>
    <shipping>
```

```
        <address>
            <street>Av. Brigadeiro Faria Lima</street>
            <number>1384</number>
            <complement>1 andar</complement>
            <district>Jardim Paulistano</district>
            <city>Sao Paulo</city>
            <state>SP</state>
            <country>BRA</country>
            <postalCode>01452002</postalCode>
        </address>
        <type>3</type>
        <cost>0.00</cost>
    </shipping>
    <creditCard>
        <token>4a56sd456a4d54asd65as4d56a4sd564</token>
        <installment>
            <quantity>2</quantity>
            <value>5.50</value>
        </installment>
        <holder>
            <name>Nome impresso no cartao</name>
            <documents>
                <document>
                    <type>CPF</type>
                    <value>00722333665</value>
                </document>
            </documents>
            <birthDate>20/10/1980</birthDate>
            <phone>
                <areaCode>11</areaCode>
                <number>999991111</number>
            </phone>
        </holder>
        <billingAddress>
            <street>Av. Brigadeiro Faria Lima</street>
            <number>1384</number>
            <complement>1 andar</complement>
            <district>Jardim Paulistano</district>
            <city>Sao Paulo</city>
            <state>SP</state>
            <country>BRA</country>
            <postalCode>01452002</postalCode>
        </billingAddress>
    </creditCard>
</payment>

```

Transparent Checkout API Parameters

The table below describes the parameters of the API of the Transparent Checkout. For integration using XML, the root element is <payment>.

PARAMETER	DESCRIPTION
HTTP Parameter: **receiverEmail** XML Element: <payment> <receiver> **<email>**	**Specifies the email that will receive the payment** **Presence:** Optional **Type:** Text **Format:** A valid e-mail, restricted to 60 characters. The email informed must be linked to the PagSeguro account that is making the call to the API.
HTTP Parameter: **notificationURL** XML Element: <payment> **<notificationURL>**	**URL for the sending of notifications.** **Presence:** Optional **Type:** Text **Format:** A valid URL, restricted to 255 characters
HTTP Parameter: **currency** XML Element: <payment> **<currency>**	**Currency used.** Indicates the currency in which the payment will be made. At this time, the only option available is **BRL** (Real). **Presence:** Required **Type:** Text **Format:** Only the BRL value is accepted
HTTP Parameter: **paymentMethod** XML Element: <payment> **<method>**	**Payment means** **Presence:** Required **Type:** Text **Format: CreditCard, "boleto"** bank payment slip or **EFT.**
HTTP Parameter: **paymentMode** XML Element: <payment> **<mode>**	**Payment method** **Presence:** Required **Type:** Text **Format:** accepts the default option.

HTTP Parameter: **itemId1, itemId2, etc.** XML Element: <checkout> <items> <item> **<id>**	**Identifier of items.** 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. **Presence:** Required **Type:** Text **Format:** Free-form, restricted to 100 characters.
HTTP Parameter: **itemDescription1, itemDescription2, etc.** XML Element: <payment> <items> <item> **<description>**	**Description of items.** 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. **Presence:** Required **Type:** Text **Format:** Free-form, restricted to 100 characters.
HTTP Parameter: **itemAmount1, itemAmount2, etc.** XML Element: <checkout> <items> <item> **<amount>**	**Unit value of items.** 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. **Presence:** Required **Type:** Number **Format:** Decimal, with two decimal places separated by a period (ex., 1234,56), greater than 0.00 and less or equal to 9999999.00.
HTTP Parameter: **itemQuantity1, itemQuantity2, etc.** XML Element: <payment> <items> <item> **<quantity>**	**Quantity of items.** 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. **Presence:** Required **Type:** Number **Format:** A whole number greater or equal to 1 and lower or equal to 999.

HTTP Parameter: **reference** XML Element: <payment> <reference>	**Reference code.** 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. **Presence:** Optional **Type:** Text **Format:** Free-form, restricted to 200 characters.
HTTP Parameter: **extraAmount** XML Element: <payment> <extraAmount>	**Additional value to be charged.** **Presence:** Optional **Type:** Number **Format:** Decimal, with two decimal places separated by a period (ex., 1234,56), greater than 0.00 and less or equal to 9999999.00.
Elemento XML: <payment> <sender>	**Buyer's data.**
HTTP Parameter: **senderEmail** XML Element: <payment> <sender> <email>	**Buyer's email.** Specifies the email of the buyer who is making the payment. **Presence:** Required **Type:** Text **Format:** A valid email (ex., user@site.com.br), restricted to 60 characters.
HTTP Parameter: **senderName** XML Element: <payment> <sender> <name>	**Buyer's full name.** Specifies the full name of the buyer who is making the payment. **Presence:** Required **Type:** Text **Format:** At least two strings of characters, restricted to a total of 50 characters.
XML Element: <payment> <sender> <documents>	**List of buyer's documents (such a ID numbers).**

XML Element: <payment> <sender> <documents> **<document>**	**Represents a buyer's document.**
XML Element: <payment> <sender> <documents> <document> **<type>**	**Type of buyer's document.** 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. **Presence:** Required **Type:** Text **Format:** Case sensitive. Only the CPF number is accepted.
HTTP Parameter: **senderCPF** XML Element: <payment> <sender> <documents> <document> **<value>**	**CPF number of owner of buyer** **Presence:** Required **Type:** Number **Format:** 11 digits.
HTTP Parameter: **senderAreaCode** XML Element: <payment> <sender> <phone> **<areaCode>**	**Area code (DDD) of buyer.** Specifies the area code (DDD) of buyer who is making the payment **Presence:** Required **Type:** Number **Format:** A 2-digit number that represents a valid area code.
HTTP Parameter: **senderPhone** XML Element: <payment> <sender> <phone> **<number>**	**Telephone number of buyer.** Specifies the telephone number of buyer who is making the payment **Presence:** Required **Type:** Number **Format:** A number comprised of 7 to 9 digits.

HTTP Parameter: **senderHash** XML Element: <payment> <sender> **<hash>**	**Customer ID.** Identifier generated by the API JavaScript of PagSeguro that identifies the Customer (hash). **Presence:** Required **Type:** Text **Format:** Obtained from a JavaScript PagseguroDirectPayment.getSenderHash() call.
XML Element: <payment> **<shipping>**	**Delivery Data**
HTTP Parameter: **shippingType** XML Element: <payment> <shipping> **<type>**	**Shipping method of product.** **Presence:** Optional **Type:** Number **Format:** 1-PAC, 2–SEDEX, 3-Unknown
HTTP Parameter: **shippingCost** XML Element: <payment> <shipping> **<cost>**	**Shipping cost** **Presence:** Optional **Type:** Number **Format:** Decimal, with two decimal places separated by a period (ex., 1234.56), greater than 0.00 and less or equal to 9999999.00.
XML Element: <payment> <shipping> **<address>**	**Shipping address data**
HTTP Parameter: **shippingAddressCountry** XML Element: <payment> <shipping> <address> **<country>**	**Country of shipping address** Specifies the country of the shipping address of the order. **Presence:** Required **Type:** Text **Format:** At this time, only the **BRA** value is allowed.

HTTP Parameter: **shippingAddressState** XML Element: <payment> <shipping> <address> **<state>**	**State of shipping address** Specifies the state of the shipping address of the product. **Presence:** Required **Type:** Text **Format:** Two letters, representing the abbreviation of the corresponding Brazilian state.
HTTP Parameter: **shippingAddressCity** XML Element: <payment> <shipping> <address> **<city>**	**City of shipping address** Specifies the city of the shipping address of the product. **Presence:** Required **Type:** Text **Format:** Free-form. Must be a valid Brazilian city name with at least 2 and no more than 60 characters.
HTTP Parameter: **shippingAddressPostalCode** XML Element: <payment> <shipping> <address> **<postalCode>**	**Postal code of shipping address** Specifies the postal code of the shipping address of the product. **Presence:** Required **Type:** Number **Format:** An 8-digit number.
HTTP Parameter: **shippingAddressDistrict** XML Element: <payment> <shipping> <address> **<district>**	**District of shipping address** Specifies the district of the shipping address of the product. **Presence:** Required **Type:** Text **Format:** Free-form, restricted to 60 characters.
HTTP Parameter: **shippingAddressStreet** XML Element: <payment> <shipping> <address> **<street>**	**Street name of shipping address** Specifies the street name of the shipping address of the product. **Presence:** Required **Type:** Text **Format:** Free-form, restricted to 80 characters.

HTTP Parameter: **shippingAddressNumber** XML Element: <payment> <shipping> <address> **<number>**	**Number of shipping address.** Specifies the number of the shipping address of the product. **Presence:** Required **Type:** Text **Format:** Free-form, restricted to 20 characters.
HTTP Parameter: **shippingAddressComplement** XML Element: <payment> <shipping> <address> **<complement>**	**Complement of shipping address.** Specifies the complement (block, apartment, etc.) of the shipping address of the product. **Presence:** Optional **Type:** Text **Format:** Free-form, restricted to 40 characters.
Data for Online Debit	
XML Element: <payment> <bank>	**Bank data.**
HTTP Parameter: **bankName** XML Element: <payment> <bank> **<name>**	**Bank name** Name of bank to which the redirect link will be generated. **Presence:** Required for Online Debit **Type:** Text **Format: bradesco, itau, bancodobrasil, banrisul or hsbc**
Card data	
HTTP Parameter: **creditCardToken** XML Element: <payment> <creditCard> **<token>**	**Credit Card Token** Token returned in the service for obtainment of the credit card token (page 9). **Presence:** Required for Credit Card **Type:** Text **Format:** Unrestricted amount of characters

HTTP Parameter: **installmentQuantity** XML Element: <payment> <creditCard> <installment> **<quantity>**	**Amount of installments** Amount of installments chosen by customer. **Presence:** Required for Credit Card **Type:** Full **Format:** Whole number
HTTP Parameter: **installmentValue** XML Element: <payment> <creditCard> <installment> **<value>**	**Value of installments** Value of installments obtained in the installment option service (page 10). **Presence:** Required for Credit Card **Type:** Number **Format:** Numeric with 2 decimal places and separated by a period. Ex.: 1111.11
Credit card holder data	
HTTP Parameter: **creditCardHolderName** XML Element: <payment> <creditCard> <holder> **<name>**	**Name printed on credit card** **Presence:** Required for Credit Card **Type:** Text **Format: min =** 1, max = 50 characters
HTTP Parameter: **creditCardHolderBirthDate** XML Element: <payment> <creditCard> <holder> **<birthDate>**	**Birth date of credit card holder** **Presence:** Required for Credit Card **Type:** dd/MM/yyyy **Format:** 31/12/2013
XML Element: <payment> <creditCard> <holder> **<documents>**	**List of documents of credit card holder.**

XML Element: <payment> <creditCard> <holder> <documents> **<document>**	**Represents a document of credit card holder.**
XML Element: <payment> <creditCard> <holder> <documents> <document> **<type>**	**Type of document of credit card holder.** Specifies the type of document of credit card holder that is being used to make the payment. **Presence:** Required for Credit Card **Type:** Text **Format:** Case sensitive. Only the **CPF** number is accepted.
HTTP Parameter: **creditCardHolderCPF** XML Element: <payment> <creditCard> <holder> <documents> <document> **<value>**	**CPF number of credit card holder** **Presence:** Required for Credit Card **Type:** Text
XML Element: <payment> <creditCard> <holder> **<phone>**	**Telephone number of credit card holder.**
HTTP Parameter: **creditCardHolderAreaCode** XML Element: <payment> <creditCard> <holder> <phone> **<areaCode>**	**Area code** **Presence:** Required for Credit Card. **Type:** Number **Format:** A 2-digit number that represents a valid area code.

HTTP Parameter: **creditCardHolderPhone** XML Element: <payment> <creditCard> <holder> <phone> **<number>**	**Telephone** **Presence:** Required for Credit Card. **Type:** Numeric **Format:** A number comprised of 7 to 9 digits.
	Billing address data
XML Element: <payment> <creditCard> **<billingAddress>**	**Billing address.**
HTTP Parameter: **billingAddressPostalCode** XML Element: <payment> <creditCard> <billingAddress> **<postalCode>**	**Postal code of billing address** **Presence:** Required for Credit Card. **Format:** An 8-digit number that represents a valid postal code (ex., 01452002).
HTTP Parameter: **billingAddressStreet** XML Element: <payment> <creditCard> <billingAddress> **<street>**	**Street Name** **Presence:** Required for Credit Card. **Format:** Free-form
HTTP Parameter: **billingAddressNumber** XML Element: <payment> <creditCard> <billingAddress> **<number>**	**Number** **Presence:** Required for Credit Card. **Format:** Free-form

HTTP Parameter: **billingAddressComplement** XML Element: <payment> <creditCard> <billingAddress> **<complement>**	**Complement** **Presence:** Required for Credit Card. **Format:** Free-form
HTTP Parameter: **billingAddressDistrict** XML Element: <payment> <creditCard> <billingAddress> **<district>**	**District** **Presence:** Required for Credit Card. **Type:** Text **Format:** Free-form, restricted to 60 characters.
HTTP Parameter: **billingAddressCity** XML Element: <payment> <creditCard> <billingAddress> **<city>**	**City** **Presence:** Required for Credit Card. **Format:** Must be a valid Brazilian city name with at least 2 and no more than 60 characters.
HTTP Parameter: **billingAddressState** XML Element: <payment> <creditCard> <billingAddress> **<state>**	**State** **Attendance:** Required for Credit Card. **Format:** Two letters representing the initials of the Brazilian state corresponding (eg, SP).
HTTP Parameter: **billingAddressCountry** XML Element: <payment> <creditCard> <billingAddress> **<country>**	**Country** **Attendance:** Required for Credit Card. **Format:** Recognizes only the BRA value.

Returno f Transparent Checkout API

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

```xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<transaction>
    <date>2011-02-05T15:46:12.000-02:00</date>
    <lastEventDate>2011-02-15T17:39:14.000-03:00</lastEventDate>
    <code>9E884542-81B3-4419-9A75-BCC6FB495EF1</code>
    <reference>REF1234</reference>
    <type>1</type>
    <status>3</status>
    <paymentMethod>
        <type>1</type>
        <code>101</code>
    </paymentMethod>
<paymentLink>
https://pagseguro.uol.com.br/checkout/imprimeBoleto.jhtml?code=314601B208B24A5CA53260000F7BB0D0
</paymentLink>
    <grossAmount>49900.00</grossAmount>
    <discountAmount>0.00</discountAmount>
    <feeAmount>0.00</feeAmount>
    <netAmount>49900.50</netAmount>
    <extraAmount>0.00</extraAmount>
    <installmentCount>1</installmentCount>
    <itemCount>2</itemCount>
    <items>
        <item>
            <id>0001</id>
            <description>Notebook Prata</description>
            <quantity>1</quantity>
            <amount>24300.00</amount>
        </item>
        <item>
            <id>0002</id>
            <description>Notebook Rosa</description>
            <quantity>1</quantity>
            <amount>25600.00</amount>
        </item>
    </items>
    <sender>
        <name>José Comprador</name>
        <email>comprador@uol.com.br</email>
        <phone>
            <areaCode>11</areaCode>
            <number>56273440</number>
        </phone>
    </sender>
    <shipping>
        <address>
            <street>Av. Brig. Faria Lima</street>
            <number>1384</number>
            <complement>5o andar</complement>
            <district>Jardim Paulistano</district>
            <postalCode>01452002</postalCode>
            <city>Sao Paulo</city>
            <state>SP</state>
            <country>BRA</country>
        </address>
        <type>1</type>
        <cost>21.50</cost>
    </shipping>
</transaction>
```

Parameters for return of API of Transparent Checkout

The table below describes the return parameters of the API of the Transparent Checkout.

FIELD	DESCRIPTION
<transaction>	This field is the root of the XML and includes the transaction data.
<transaction> **<date>**	**Date of creation of transaction.** Specifies the time when the transaction was created. **Presence:** Required **Type:** Date/time. **Format:** YYYY-MM-DDThh:mm:ss.sTZD, the official format of W3C for dates. See more about formatting dates.
<transaction> **<lastEventDate>**	**Date of last event.** Specifies the time in which the last change in the status of the transaction took place. **Presence:** Required **Type:** Date/time. **Format:** YYYY-MM-DDThh:mm:ss.sTZD, the official format of W3C for dates. See more about formatting dates.
<transaction> **<code>**	**ID code of transaction** Returns the code that uniquely identifies the transaction. **Presence:** Required **Type:** Text. **Format:** A string of 36 characters.
<transaction> **<reference>**	**Reference code of transaction** 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. **Presence:** Optional

<table>
<tr><th>FIELD</th><th>DESCRIPTION</th></tr>
<tr><td></td><td><b>Type:</b> Text.
<b>Format:</b> Free-form, restricted to 200 characters.</td></tr>
<tr><td>&lt;transaction>
<b>&lt;type&gt;</b></td><td><b>Transaction type.</b>

Represents the type of transaction received. The more usual values for this field and their respective results are described below.

<table>
<tr><th>Code</th><th>Meaning</th></tr>
<tr><td><b>1</b></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>

Other less usual types of transactions have been omitted. Take note that new types can be added in future versions of the API.

<b>Presence:</b> Required.
<b>Type:</b> Number.
<b>Format:</b> Whole number.</td></tr>
<tr><td>&lt;transaction>
<b>&lt;status&gt;</b></td><td><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.

<table>
<tr><th>Code</th><th>Meaning</th></tr>
<tr><td><b>1</b></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><b>2</b></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><b>3</b></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><b>4</b></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>
</table>
</td></tr>
</table>

<table>
<tr><th>FIELD</th><th>DESCRIPTION</th></tr>
<tr><td></td><td>
<table>
<tr><td><b>5</b></td><td><b>Under claim:</b> the buyer, within the clearing period of the transaction, opened a claim.</td></tr>
<tr><td><b>6</b></td><td><b>Refunded:</b> the transaction value was returned to the buyer.</td></tr>
<tr><td><b>7</b></td><td><b>Canceled:</b> the transaction was canceled without having been concluded.</td></tr>
</table>
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.
<b>Presence:</b> Required.
<b>Type:</b> Number.
<b>Format:</b> Whole number.
</td></tr>
<tr><td>&lt;transaction>
<b>&lt;cancellationSource&gt;</b></td><td>
<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.
<table>
<tr><th>Code</th><th>Meaning</th></tr>
<tr><td>INTERNAL</td><td><b>PagSeguro</b></td></tr>
<tr><td>EXTERNAL</td><td><b>Financial Institutions</b></td></tr>
</table>
<b>Presence:</b> Optional (only when transactionStatus is equal to 7).
<b>Type:</b> Text.
<b>Format:</b> Possible INTERNAL or EXTERNAL values.
</td></tr>
<tr><td>&lt;transaction>
<b>&lt;paymentMethod&gt;</b></td><td>Data of payment method used by buyer.</td></tr>
<tr><td>&lt;transaction>
<paymentMethod>
<b>&lt;type&gt;</b></td><td>
<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.
</td></tr>
</table>

<table>
<tr><th>FIELD</th><th>DESCRIPTION</th></tr>
<tr><td></td><td>
<table>
<tr><th>Code</th><th>Meaning</th></tr>
<tr><td><b>1</b></td><td><b>Credit card</b>: the buyer chose to pay the transaction using a credit card.</td></tr>
<tr><td><b>2</b></td><td><b>“Boleto” payment slip:</b> the buyer chose to pay with a bank payment slip (“boleto”).</td></tr>
<tr><td><b>3</b></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><b>4</b></td><td><b>PagSeguro Balance:</b> the buyer chose to pay for the transaction using the balance of his/her PagSeguro account.</td></tr>
<tr><td><b>5</b></td><td><b>Oi Paggo</b>*: the buyer chose to pay the transaction through his/her Oi mobile phone.</td></tr>
<tr><td><b>7</b></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>
<tr><td colspan="2">* The marked types are not available for use.</td></tr>
</table>

<b>Presence:</b> Required.
<b>Type:</b> Number.
<b>Format:</b> Whole number.</td></tr>
<tr><td>&lt;transaction>
<b>&lt;paymentLink&gt;</b></td><td><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.

<b>Presence:</b> Only for payments via “Boleto” and TEF.
<b>Type:</b> Text.
<b>Format:</b> URL</td></tr>
<tr><td>&lt;transaction>
<paymentMethod>
<b>&lt;code&gt;</b></td><td><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</td></tr>
</table>

<table>
<tr><th>FIELD</th><th>DESCRIPTION</th></tr>
<tr><td></td><td>used or the bank chosen for online debit. The table below describes the possible values and their meanings.
<table>
<tr><th>Code</th><th>Meaning</th></tr>
<tr><td>101</td><td><b>Visa credit card.</b></td></tr>
<tr><td>102</td><td><b>Mastercard credit card.</b></td></tr>
<tr><td>103</td><td><b>American Express credit card.</b></td></tr>
<tr><td>104</td><td><b>Diners credit card.</b></td></tr>
<tr><td>105</td><td><b>Hipercard credit card.</b></td></tr>
<tr><td>106</td><td><b>Aura credit card.</b></td></tr>
<tr><td>107</td><td><b>Elo credit card.</b></td></tr>
<tr><td>108</td><td><b>PLENOcard credit card.</b></td></tr>
<tr><td>109</td><td><b>PersonalCard credit card.</b></td></tr>
<tr><td>110</td><td><b>JCB credit card.</b></td></tr>
<tr><td>111</td><td><b>Discover credit card.</b></td></tr>
<tr><td>112</td><td><b>BrasilCard credit card.</b></td></tr>
<tr><td>113</td><td><b>FORTBRASIL credit card.</b></td></tr>
<tr><td>114</td><td><b>CARDBAN credit card.</b></td></tr>
<tr><td>115</td><td><b>VALECARD credit card.</b></td></tr>
<tr><td>116</td><td><b>Cabal credit card.</b></td></tr>
<tr><td>117</td><td><b>Mais! credit card.</b></td></tr>
<tr><td>118</td><td><b>Avista credit card.</b></td></tr>
<tr><td>119</td><td><b>GRANDCARD credit card.</b></td></tr>
<tr><td>201</td><td><b>Bradesco “boleto”*</b></td></tr>
<tr><td>202</td><td><b>Santander “boleto”.</b></td></tr>
<tr><td>301</td><td><b>Bradesco online debit.</b></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>FIELD</th><th>DESCRIPTION</th></tr>
<tr><td></td><td>
<table>
<tr><td>302</td><td>**Itau online debit.**</td></tr>
<tr><td>303</td><td>**Unibanco online debit.***</td></tr>
<tr><td>304</td><td>**Banco do Brasil online debit.**</td></tr>
<tr><td>305</td><td>**Banco Real online debit.***</td></tr>
<tr><td>306</td><td>**Banrisul online debit.**</td></tr>
<tr><td>307</td><td>**HSBC online debit.**</td></tr>
<tr><td>401</td><td>**PagSeguro balance.**</td></tr>
<tr><td>501</td><td>**Oi Paggo.***</td></tr>
<tr><td>701</td><td>**Deposit in account – Banco do Brasil**</td></tr>
<tr><td colspan="2">* The marked payment methods are not available for use.</td></tr>
</table>

Presence: Required.
Type: Number.
Format: Whole number.</td></tr>
<tr><td>&lt;transaction>
<grossAmount&gt;</td><td>**Gross value of transaction.**

Specifies the gross value of the transaction, calculated by adding the prices of all the items in the payment.

Presence: Required.
Type: Number.
Format: Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.</td></tr>
<tr><td>&lt;transaction>
<discountAmount&gt;</td><td>**Value of discount granted.**

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.

Presence: Required.
Type: Number.
Format: Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.</td></tr>
</table>

FIELD	DESCRIPTION
<transaction> **<feeAmount>**	**Total amount of charged fees.** Specifies the total value of fees charged by PagSeguro in this transaction. **Presence:** Required. **Type:** Number. **Format:** Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.
<transaction> **<netAmount>**	**Net value of transaction.** 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. **Presence:** Required. **Type:** Number. **Format:** Decimal, with two decimal places separated by a period (“.”). For example, 1234.56**.**
<transaction> **<escrowEndDate>**	**Credit date.** Date in which the transaction value will be available in the seller’s account. **Presence:** Present only when the transaction status is one of the following values: Paid (3), Available (4), Under claim (5) or Returned (6). **Type:** Date/Time. **Format:** YYYY-MM-DDThh:mm:ss.sTZD, the official format of W3C for dates. See more about formatting dates.
<transaction> **<extraAmount>**	**Extra amount.** 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. **Presence:** Required. **Type:** Number. **Format:** Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.

FIELD	DESCRIPTION
<transaction> **<installmentCount>**	**Amount of installments.** Indicates the number of installments that the buyer chose in the payment using a credit card. **Presence:** Required. **Type:** Number. **Format:** Whole number.
<transaction> **<itemCount>**	**Number of items in transaction.** Indicates out the number of items contained in this transaction. **Presence:** Required. **Type:** Number. **Format:** Whole number.
<transaction> **<items>**	List of items contained in the transaction. The number of items under this item represent the value of the item count.
<transaction> <items> **<item>**	Represents an item of the transaction.
<transaction> <items> <item> **<id>**	**ID of item.** Identifies the item of the transaction. This identifier must be unique per transaction and must be informed in the payment flow. **Presence:** Required. **Type:** Text. **Format:** Free-form.
<transaction> <items> <item> **<description>**	**Item description.** Describes a single item in the transaction. The same description that was provided during the payment flow. **Presence:** Required. **Type:** Text. **Format:** Free-form.
<transaction> <items> <item>	**Unit value of item.** Specifies the unit price of the item of the transaction. This is the value that was specified in the payment flow.

FIELD	DESCRIPTION
<amount>	**Presence:** Required. **Type:** Number. **Format:** Decimal, with two decimal places separated by a period (“.”). For example, 1234.56.
<transaction> <items> <item> **<quantity>**	**Quantity of item.** Specifies the quantity of the item in the transaction. This is the quantity that was specified in the payment flow. **Presence:** Required. **Type:** Number. **Format:** A whole number greater or equal to 1 and less or equal to 999.
<transaction> **<sender>**	Buyer data.
<transaction> <sender> **<email>**	**Buyer’s email.** Specifies the email of the buyer who made the transaction **Presence:** Required. **Type:** Text. **Format:** A valid email (ex., user@site.com.br), restricted to 60 characters.
<transaction> <sender> **<name>**	**Buyer’s full name.** Specifies the full name of the buyer who made the payment. **Presence:** Optional **Type:** Text **Format:** At least two strings of characters, restricted to a total of 50 characters.
<transaction> <sender> **<phone>**	Buyer telephone data
<transaction> <sender> <phone>	**Area code (DDD) of buyer.** Specifies the area code (DDD) of buyer who made the payment

<table>
<tr><th>FIELD</th><th>DESCRIPTION</th></tr>
<tr><td><b>&lt;areaCode&gt;</b></td><td><b>Presence:</b> Optional
<b>Type:</b> Number
<b>Format:</b> A 2-digit number that represents a valid area code.</td></tr>
<tr><td>&lt;transaction>
<sender>
<phone&gt;
<b>&lt;number&gt;</b></td><td><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.</td></tr>
<tr><td>&lt;transaction>
<b>&lt;shipping&gt;</b></td><td>Shipping data.</td></tr>
<tr><td>&lt;transaction>
<shipping>
<b>&lt;type&gt;</b></td><td><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>
<tr><th>Code</th><th>Meaning</th></tr>
<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>
</table>
<b>Presence:</b> Required.
<b>Type:</b> Number
<b>Format:</b> Whole.</td></tr>
<tr><td>&lt;transaction>
<shipping>
<b>&lt;cost&gt;</b></td><td><b>Total shipping cost.</b>
Specifies the total shipping cost, based on the shipping options informed 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.</td></tr>
</table>

FIELD	DESCRIPTION
<transaction> <shipping> **<address>**	Shipping address data.
<transaction> <shipping> <address> **<country>**	**Country of shipping address** Specifies the country of the shipping address of the product. **Presence:** Optional **Type:** Text **Format:** At this time, only the **BRA** value is allowed.
<transaction> <shipping> <address> **<state>**	**State of shipping address** Specifies the state of the shipping address of the product. **Presence:** Optional **Type:** Text **Format:** Two letters, representing the abbreviation of the corresponding Brazilian state.
<transaction> <shipping> <address> **<city>**	**City of shipping address** Specifies the city of the shipping address of the product. **Presence:** Optional **Type:** Text **Format:** Free-form. Must be a valid Brazilian city name, in accordance with Post Office data.
<transaction> <shipping> <address> **<postalCode>**	**Postal code of shipping address** Specifies the postal code of the shipping address of the product. **Presence:** Optional **Type:** Number **Format:** An 8-digit number.
<transaction> <shipping> <address> **<district>**	**District of shipping address** Specifies the district of the shipping address of the product. **Presence:** Optional **Type:** Text **Format:** Free-form.

FIELD	DESCRIPTION
<transaction> <shipping> <address> **<street>**	**Street name of shipping address** Specifies the street name of the shipping address of the product. **Presence:** Optional **Type:** Text **Format:** Free-form.
<transaction> <shipping> <address> **<number>**	**Number of shipping address.** Specifies the number of the shipping address of the product. **Presence:** Optional **Type:** Text **Format:** Free-form.
<transaction> <shipping> <address> **<complement>**	**Complement of shipping address.** Specifies the complement (block, apartment, etc.) of the shipping address of the product. **Presence:** Optional **Type:** Text **Format:** Free-form.

Error codes

See below the list of errors that PagSeguor may return to your application:

CÓDIGO	DESCRIÇÃO
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}

CÓDIGO	DESCRIÇÃO
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.
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.

CÓDIGO	DESCRIÇÃO
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.
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}

CÓDIGO	DESCRIÇÃO
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}
53077	item amount is required.
53078	item amount invalid pattern: {0}. Must fit the patern: \\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}

CÓDIGO	DESCRIÇÃO
53097	shipping cost out of range: {0}
53098	cart total value is negative: {0}
53099	extra amount invalid pattern: {0}. Must fit the patern: -?\\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}
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.

⚠ **Warning:** PagSeguro may include new error messages if necessary without prior notice.