Content

INTRODUCTION

PrimeiroPay Integration Guide
Copy and Pay – Hosted Payment Page
Version 07.2016

Definitions:

Server-to-Server: You will use your own frontend to build the checkout for your customers

Copy and Pay: Is our hosted Payment Page in a *PCI DSS level 1* compliant environment

For this guide all transactions are POST transactions and should be targeted at:

LIVE: https://oppwa.com/v1/payments

TEST: https://test.oppwa.com/v1/payments

Please observe the following information for **BRAZIL**:

- **Boleto payments** are a server-to-server <u>only</u> payment method and follow the asynchronous workflow which can be found in the second section of this page:
  https://primeiropay.docs.oppwa.com/tutorials/server-to-server

This flow is solely to create / issue the Boleto.

In order to retrieve all paid Boletos, you will need to query PrimeiroPay to retrieve all paid Boletos since Boletos take at least 24 hours to be processed by the bank. Please refer to the Boleto section in this document for further instructions.

The collection of the CPF (Tax Identification number) is mandatory for all payments in Brazil. <u>It is also mandatory to verify the CPF via this Java Script</u>: https://gist.github.com/cauerego/873308

Please observe the following information for **IRAN**:

Shetab is an asynchronous workflow available as a non server-to-server payment method, however, the last step "Get the payment status" is the final payment status whether the transaction was successful or not. You will not need to send any additional query.

2 Payments in Brazil

2.1 Boleto Payments via server-to-server

Server-to-Server Integration Example:
(All these fields are required to process the payment)

amount=1.00
currency=BRL
paymentBrand=BOLETO
paymentType=PA
authentication.userId=8a8294174ae82ada014aedf7fb151d5f
authentication.password=AKdx7nmH
authentication.entityId=8a8294174ae82ada014aedfd75aa1d79
customer.givenName=Pagador
customer.surname=Pagador
customer.identificationDocType=TAXSTATEMENT
billing.street1=Av Brigadeiro Faria Lima 3729
billing.city=Sao Paulo
billing.state=SP
billing.postcode=04538905
billing.country=BR
customParameters[BRADESCO_cpfsacado]=11111111111
customer.ip=1.1.1.1
customer.email=test@test.com
customer.identificationDocId=customParameters[BRADESCO_cpfsacado]
customParameters[CUSTOM_due_date]=01012016
merchantTransactionId=1234
testMode=EXTERNAL

Obs: all *purple* fields are fixed values or in case of customer data can be filled with the example / dummy data above. All other fields have to be collecte during the payment process.
It is recommended that the customer.email field is filled by the accountID of the game/software/service. Please add "@yourchoice.com" behind the accountID.
testMode=External is only when you send the transaction to the Test Environment and will connect to the bank.

Please replace all yellow marked fields with your credentials.

Field specifications of all fields to be filled in by the Merchant:

Name	Description	Format	Example
amount	Amount of the payment request. The dot is used as a decimal separator.	N13 [0-9]{1,10}\.[0-9]{2}	1.00
currency	Currency Code according to ISO 4217 specifications of the payment request's amount	A3 [a-zA-Z]{3}	BRL
paymentBrand	The brand specifies the method for the request	AN32 [a-zA-Z0- 9_] {1,32}	BOLETO
paymentType	The payment type for the request	A2	PA
Authentication.userID	The userId for the entity. Required for a server-to-server request	AN32 [a-zA-Z0- 9]{32}	8a8294174ae82ada014 aedf7fb151d5 f
Authentication.password	The password for the entity (channel). Required for a server-to-server request	AN32 [a-zA-Z0- 9]{8,32}	AKdx7nmH
authentication.entityId	The entity for the request. (e.g. channel)	AN32 [a-zA-Z0- 9]{32}	8a8294174ae82ada014 aedfd75aa1d79
customer.email	The (internal) email address of the customer	AN128 [\s\S]{6,128}	test@test.com
customer.ip	AN255 [\s\S]{1,255}	AN255 [\s\S]{1,255}	1.1.1.1
customParameters[BRADESCO_cpfsacado]	CPF number of customer	N11	70383053102
customer.identificationDocId	CPF number of customer	N11	70383053102

merchantTransactionID	Merchant-provided reference number	AN255 [\s\S]{1,255}	1234
customParameters[CUSTOM_due_date]	Due Date of the Boleto. Recommended: 3 days in the future.	AN10 {19|20}([0-9]{2})-(0[1-9]|1[0-2])-(0[1-9]|1[0-9]|2[0-9]|3[0-1])	01012016

2.1.1 Re-Direct the shopper

The next step is to redirect the account holder. To do this you must parse the `redirect.url` from the Initial Payment response along with any parameters. If parameters are present they should be POST in the redirect, otherwise a straight forward redirect to the `redirect.url` is sufficient.

```
<form action="{shopperResultUrl}" class="paymentWidgets">

  <input type="text" name="{name-1}" value="{value-1}">
  ...
  <input type="text" name="{name-2}" value="{value-2}">

</form>
```

After a successful re-direction you will see a blank page with a banking header. On Production Environment you will see the actual Boleto:

Since the re-direction is to the bank, the user will not be forwarded to any success page, hence you will not need to define a success page.

2.1.2 Retrieve paid Boletos

Automatically retrieve if somebody has successfully paid the Boleto. Receipts need to be retrieved in order to confirm if a prepayment has been fulfilled.

Test-URL: https://test.ctpe.io/payment/query
Live-URL: https://ctpe.io/payment/query

```
<Request version="1.0">
<Security sender="123a456b789c123d456e789f012g345"/>
</Header>
<Query mode="LIVE" level="CHANNEL"
entity="678a456b789c123d456e789f012g432"
type="STANDARD">
<Period from="2004-09-01" to="2004-09-01"/>
<Types>
<Type code="RC"/>
</Types>
</Query>
</Request>
```

Query mode on Production environment is LIVE
Query mode on Test Environment is "CONNECTOR_TEST"

This request will retrieve the receipts for the channel identified by the ID 678a456b789c123d456e789f012g432 for day of first of September 2014.

We recommend to run the script once a day at 1pm UTC. To ensure all paid Boletos will be reported back to you, we recommend to set the period from the previous day to the current day as some Boletos may be confirmed manually after 1pm UTC to our system.

Assuming one receipt has been issued during the aforementioned period, the following will be returned:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response version="1.0">
    <Transaction mode="LIVE"
channel="678a456b789c123d456e789f012g432" response="SYNC">
        <Identification>
            <ShortID>9974.5474.4274</ShortID>
<UniqueID>8a83948152d0503f0152d4310e6a22e0</UniqueID>
```

```xml
            <TransactionID>1234</TransactionID>
<ShopperID>a4e6dd4587b049e0268cfef292786a007de4e4867d5cc5cc833
54a8f72c3131755b2e655d52f1434863667b6b1ed4b23f6bce25e3d849ffca
f114002a5f852ce</ShopperID>
            <InvoiceID> </InvoiceID>
<ReferenceID>8a83948752d050410152d1aa4ed155e8</ReferenceID>
</Identification>
        <Payment code="PP.RC">
            <Clearing>
                <Amount>1.00</Amount>
                <Currency>BRL</Currency>
                <Descriptor>BOLETO</Descriptor>
                <FxRate>1.0</FxRate>
                <FxSource>INTERN</FxSource>
                <FxDate>2016-01-01 06:35:30</FxDate>
            </Clearing>
        </Payment>
        <Processing code="PP.RC.90.00">
            <Timestamp>2016-02-12 06:35:30</Timestamp>
            <Result>ACK</Result>
            <Status code="90">NEW</Status>
            <Reason code="00">Successful Processing</Reason>
            <Return code="000.000.000">Transaction
succeeded</Return>
        </Processing>
    </Transaction>
</Response>
```

2.2 Card Payments

2.2.1 Prepare the Checkout

Integration Example:
(All these fields are required to process the payment)

GET Request to the following URL:
https://test.oppwa.com/v1/checkouts/

```
amount=1.00
currency=BRL
paymentType=PA
authentication.userId=8a8294174ae82ada014aedf7fb151d5f
authentication.password=AKdx7nmH
authentication.entityId=8a8294174ae82ada014aedfd75aa1d79
customer.ip=1.1.1.1
customer.email=test@test.com
merchantTransactionID=1234
```

Obs: all *purple* fields are fixed values or in case of customer data can be
filled with the example / dummy data above. All other fields have to be
collecte during the payment process.
It is recommended that the customer.email field is filled by the accountID
of the game/software/service. Please add "@yourchoice.com" behind the
accountID

paymentType=PA means Pre Authorization.

It is mandatory to capture the amount. We recommend to do so 1 minute after
the successful Pre Authorization. For some merchants it might make sense to
keep the Pre Authorization for a few hours or days. Please contact your
Account Manager for further instructions.
Please see section 2.2.4 how to capture the transaction

Once you created the checkout you will get a response like this:

```
{
  "result":{
    "code":"000.200.100",
    "description":"successfully created checkout"
```

```
  },

  "buildNumber":"beb408d1f2c902bfb8665e2b4e268f1039848d03@2016-02-16
12:49:58 +0000",

  "timestamp":"2016-02-16 13:25:09+0000",

  "ndc":"DA84DCEB01BD28810E992D9EC8085985.sbg-vm-tx01",

  "id":"DA84DCEB01BD28810E992D9EC8085985.sbg-vm-tx01"

}
```

2.2.2 Create the payment form

Integration Example:
(All these fields and scripts are required to process the payment)

To create the payment form you just need to add the following lines of HTML/Javascript to your page and populating the following variables

```
<script>
var wpwlOptions ={ showTaxNumberField:true, locale:"pt"
}
</script>
<script
src="https://test.oppwa.com/v1/paymentWidgets.js?checkoutId=75E53EDE2467B24C1795
01962555376C.sbg-vm-tx02"></script>
<form action="   {shopperResultUrl}" class="paymentWidgets">VISA MASTER ELO</form>
```

- The script in the beginning displays the CPF field and displays all fields in Portuguese language.
- The *checkout's id* that you got in the response from step 1
- The shopperResultUrl, which is the page on your site where the customer should be redirected to after the payment is processed
- VISA MASTER ELO are the supported brands in Brazil.

2.2.3 Get the Payment Status

Once the payment has been processed, the customer is redirected to your shopperResultUrl along with a GET parameter resourcePath.
Important: The baseUrl must end in a "/", e.g. "https://test.oppwa.com/".

Then, to get the status of the payment, you should make a GET request to the baseUrl + resourcePath, including your authentication parameters.

Integration Example:
(All mandatory fields to get the payment status)

GET Request to the following URL:
https://test.oppwa.com/v1/checkouts/{checkoutId}/payment
The id is what you receive in the previous response once a payment was initiated.

authentication.userId=8a8294184e736012014e78a17a6a15b0
authentication.entityId=8a8294184e736012014e78a17a5615ac
authentication.password=f2FEKZqtBz

We recommend that you verify the following fields from the Payment Status response, by comparing the returned values with expected:
- ID(s)
- Amount
- Currency
- Brand
- Type

2.2.4 Capture an amount

Integration Example to Capture an amount:

authentication.userId=8a8294184e736012014e78a17a6a15b0
authentication.entityId=8a8294184e736012014e78a17a5615ac
authentication.password=f2FEKZqtBz
amount=1.00
currency=BRL
paymentType=CP

A capture is used to request clearing for previously authorized funds. A capture request is performed against a previous preauthorization (PA) payment by referencing its payment.id and sending a POST request over HTTPS to the /payments/{id} endpoint. Captures can be for full or partial amounts and multiple capture requests against the same PA are allowed.

Example:
https://test.oppwa.com/v1/payments/8a82944952e4864b0152e69f4b474b3a

Please refer to our backoffice documentation for further instructions:

https://primeiropay.docs.oppwa.com/tutorials/manage-payments/backoffice#capture

Field specifications of all fields to be filled in by the Merchant:

Name	Description	Format	Example
amount	Amount of the payment request. The dot is used as a decimal separator.	N13 [0-9]{1,10}\.[0 -9]{2}	1.00
currency	Currency Code according to ISO 4217 specifications of the payment request's amount	A3 [a-zA-Z]{3}	BRL
customer.email	The (internal) email address of the customer	AN128 [\s\S]{6,12 8}	test@test.com
customer.ip	AN255 [\s\S]{1,25 5}	AN255 [\s\S]{1,25 5}	1.1.1.1
merchantTransactionID	Merchant-provided reference number	AN255 [\s\S]{1,25 5}	1234

2.2.5 Retrieve Chargebacks and Refunds

For retrieving chargebacks for a certain period a request has to be sent to the query engine at:

Test-URL: https://test.ctpe.io/payment/query

Live-URL: https://ctpe.io/payment/query

Typical Use Case: Automatically retrieve chargebacks from the payment system.

Depending on the acquirer or bank, new Chargebacks usually appear once a day in the system. For details contact your account manager. It is therefore recommended to run a request to retrieve all chargebacks from the **previous** day.

Please note: For system safety and stability reasons the merchant query access may be blocked in case of receiving too many automated requests from this account within a short timeframe.

Type code=CB is for Chargebacks
Type code=RF is for Refunds

```
<Request version="1.0">
<Security sender="123a456b789c123d456e789f012g345"/>
</Header>
<Query mode="LIVE" level="CHANNEL"
entity="678a456b789c123d456e789f012g432"
type="STANDARD">
<Period from="2004-09-01" to="2004-09-01"/>
<Types>
<Type code="CB"/>
</Types>
</Query>
</Request>
```

This request will retrieve the chargebacks for the channel identified by the ID 678a456b789c123d456e789f012g432 for the day of 1st of September. Assuming two chargebacks have been issued during the aforementioned period, the following will be returned:

```
<Response version="1.0">
<Result response="SYNC" type="LIST">
<Transaction mode="LIVE" response="SYNC"
channel="678a456b789c123d456e789f012g432">
```

```xml
<Identification>
<TransactionID>4711</TransactionID>
<UniqueID>h987i654j321k098l765m43>
<ShortID>1234.5678.9876</ShortID>
<ReferenceID>m123n456o789p876q543r210s123t456</ReferenceID>
</Identification>
<Processing code="DD.CB.90.00">
    <Timestamp>2014-11-13 08:19:52</Timestamp>
    <Result>ACK</Result>
    <Status code="90">NEW</Status>
    <Reason code="00">Successful Processing</Reason>
    <Return code="000.100.206">Revocation or Dispute</Return>
<Result>ACK</Result>
<Status code="90">NEW</Status>
<Reason code="00">Successful Processing</Reason>
<Return code="000.100.206">Revocation or Dispute</Return>
</Processing>
<Account><Number>*****6581</Number>
<Bin>456789</Bin>
<Holder>bob kosel</Holder>
<Bank>38050000</Bank>
<Country>DE</Country>
</Account>
<Payment code="DD.CB">
<Presentation>
<Amount>1.00</Amount>
<Currency>EUR</Currency>
</Presentation>
<Clearing>
<Amount>1.00</Amount>
<Currency>EUR</Currency>
<Date>2004-09-02</Date>
</Clearing>
</Payment>
</Transaction>
<Transaction mode="LIVE" response="SYNC"
channel="678a456b789c123d456e789f012g432">
<Identification>
<TransactionID>4712</TransactionID>
<UniqueID>h987i654j321k098l765m43333222111</UniqueID>
<ShortID>1234.5678.8191</ShortID>
<ReferenceID>m123n456o789p876q543r21111222333</ReferenceID>
</Identification>
<Processing code="DD.CB.00.00">
<Timestamp>2004-09-07 14:58:07</Timestamp>
<Result>ACK</Result>
<Status code="90">NEW</Status>
```

```xml
<Reason code="00">Successful Processing</Reason>
<Return code="000.100.206">Revocation or Dispute</Return>
</Processing>
<Account>
<Number>*****6581</Number>
<Bin>456789</Bin>
<Holder>bob kosel</Holder>
<Bank>38050000</Bank>
<Country>DE</Country>
</Account>
<Payment code="DD.CB">
<Presentation>
<Amount>3.00</Amount>
<Currency>EUR</Currency>
</Presentation>
<Clearing>
     <Amount>3.00</Amount>
     <Currency>EUR</Currency>
     <Date>2004-09-07</Date>
</Clearing>
</Payment>
</Transaction>
</Result>
</Reponse>
```

2.3 Paysafecard

2.3.1 Prepare the Checkout

Integration Example:
(All these fields are required to process the payment)

GET Request to the following URL:
https://test.oppwa.com/v1/checkouts/

```
amount=1.00
currency=BRL
paymentType=PA
authentication.userId=8a8294174ae82ada014aedf7fb151d5f
authentication.password=AKdx7nmH
authentication.entityId=8a8294174ae82ada014aedfd75aa1d79
customer.merchantCustomerId=7822446
shopperResultUrl=http://merchant.com/success.php
customer.email=test@test.com
merchantTransactionId=1234
```

Obs: all *purple* fields are fixed values or in case of customer data can be filled with the example / dummy data above. All other fields have to be collecte during the payment process.

Once you created the checkout you will get a response like this:

```
{

  "result":{

    "code":"000.200.100",

    "description":"successfully created checkout"

  },

  "buildNumber":"beb408d1f2c902bfb8665e2b4e268f1039848d03@2016-02-16
12:49:58 +0000",

  "timestamp":"2016-02-16 13:25:09+0000",

  "ndc":"DA84DCEB01BD28810E992D9EC8085985.sbg-vm-tx01",

  "id":"DA84DCEB01BD28810E992D9EC8085985.sbg-vm-tx01"

}
```

2.3.2 Create the payment form

Integration Example:
(All these fields and scripts are required to process the payment)

To create the payment form you just need to add the following lines of HTML/Javascript to your page and populating the following variables

```
<script>
var wpwlOptions ={ showTaxNumberField:true, locale:"pt"
}
</script>
<script
src="https://test.oppwa.com/v1/paymentWidgets.js?checkoutId=75E53EDE2467B24C1795
01962555376C.sbg-vm-tx02"></script>
<form action="  {shopperResultUrl}" class="paymentWidgets">PAYSAFECARD</form>
```

- The script in the beginning displays the CPF field and displays all fields in Portuguese language.
- The *checkout's id* that you got in the response from step 1
- The shopperResultUrl, which is the page on your site where the customer should be redirected to after the payment is processed
- PAYSAFECARD is the trigger to re-direct the user to PaysafeCard.

Once you enter a CPF number (example: 70383053102) you will be re-directed to Paysafecard. Use this code to complete the payment: *0000 0000 0990 1851*

2.3.3 Get the Payment Status

Once the payment has been processed, the customer is redirected to your shopperResultUrl along with a GET parameter resourcePath.
Important: The baseUrl must end in a "/", e.g. "https://test.oppwa.com/".
Then, to get the status of the payment, you should make a GET request to the baseUrl + resourcePath, including your authentication parameters.

Integration Example:
(All mandatory fields to get the payment status)

GET Request to the following URL:
https://test.oppwa.com/v1/checkouts/{checkoutId}/payment
The id is what you receive in the previous response once a payment was initiated.

```
authentication.userId=8a8294184e736012014e78a17a6a15b0
authentication.entityId=8a8294184e736012014e78a17a5615ac
authentication.password=f2FEKZqtBz
```

We recommend that you verify the following fields from the Payment Status response, by comparing the returned values with expected:
- ID(s)
- Amount
- Currency
- Brand
- Type

Field specifications of all fields to be filled in by the Merchant:

Name	Description	Format	Example
amount	Amount of the payment request. The dot is used as a decimal separator.	N13 [0-9]{1,10}\.[0-9]{2}	1.00

currency	Currency Code according to ISO 4217 specifications of the payment request's amount	A3 [a-zA-Z]{3}	BRL
customer.email	The (internal) email address of the customer	AN128 [\s\S]{6,128}	test@test.com
merchantTransactionID	Merchant-provided reference number	AN255 [\s\S]{1,255}	1234
customer.merchantCustomerId	A unique identifier of the shopper (the unique ID of the shopper as registered at the merchant's database)	AN32 [a-zA-Z0- 9]{50}	Client123
shopperResultUrl	Result URL after payment is completet	AN2048 [\s\S]{6,2048}	http://merchant.com/success.php

3 Result Codes

Please find a complete list of Result Codes here:
https://primeiropay.docs.oppwa.com/reference/resultCodes

4 Support

Please contact your Account Manager for any inquiries.

You can reach our Tech Support team at tech-support@primeiropay.com