PayPal Plus

Integration Guide

This document is confidential and its distribution is restricted.

DISTRIBUTE THIS DOCUMENT ONLY UNDER PROPER NDA.

Publication Date:
09/06/16

PayPal

Document Revision History

Pub. Date	Description of Changes
03/24/15	Initial release of document.
04/10/15	Updated to include creating experience profile
06/18/15	Removed shipping address as an optional parameter for the config.
07/14/15	Updated to include callback for handling errors.
07/17/15	Removed references to buttonLocation configuration parameter.
07/20/15	Clarified requirements of payerPhone, payerTaxId, and payerTaxIdType.
07/28/15	Updated JavaScript URL location
30/06/16	Major document update, payment review section, execute payment API and iFrame errors handler, experience profile ID examples.
14/11/16	Included explanation on how to create a sandbox account and generating it's REST API credentials on the developer portal.
06/02/16	New structure, iFrame javascript and listener updated.
08/24/17	New iFrame javascript errors & description added.
10/10/17	iFrame Javascript sample code updated
13/11/17	Removing Experience Profile ID API & providing listener JS alerts

Notice of non-liability: PayPal, Inc. is providing the information in this document to you "AS-IS" with all faults. PayPal, Inc. makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. PayPal, Inc. assumes no liability for damages (whether direct or indirect), caused by errors or omissions, or resulting from the use of this document or the information contained in this document or resulting from the application or use of the product or service described herein. PayPal, Inc. reserves the right to make changes to any information herein without further notice. PayPal Inc. does not guarantee that the features described in this document will be announced or made available to anyone in the future.

Table of Contents

Chapter 1. Introduction 4
PayPal Plus features 4
How PayPal Plus works 4
Chapter 2. Before you begin 5
PayPal Sandbox 5
Creating a Merchant sandbox account 5
Generating API Credentials 9
Environments & API Endpoints 12
Chapter 3. Building a payment solution with PayPal Plus 13
Overview 13
Payment flow with PayPal Plus 14
Integrating with PayPal Plus 14
Requesting an access token 14
Creating a payment 16
Integrating the Payment Selection page 19
Paypal Plus Javascript library 19
HTML DIV Placeholder 19
Paypal Plus Javascript object 19
Sample Payment Selection Pages 23
Integrating the Javascript iFrame listener 24
Handling errors on the iFrame 29
Executing the payment 30
Execute payment API response 32
Handling errors on the execute payment API 33
Payment review 34
Chapter 4. Useful REST API operations 36

Chapter 1. Introduction

PayPal Plus provides a flexible credit-card processing solution for merchants as described in the following sections.

PayPal Plus features

With PayPal Plus, merchants can build a payment processing solution where costumers are not required to have a PayPal account. Additional features include the following:

- Support for payment installments
- Ability to remember up to three credit cards *(Also remember the CVV)*
- Seller protection *(For tangible items)*
- Buyer protection *(Only for customers that create a PayPal account with the email sent from PayPal after a successful transaction within 45 days.)*

How PayPal Plus works

With PayPal Plus, your customers will be able to pay without a PayPal account only using their credit cards, PayPal Plus is a hosted payment solution that will be included at your payment selection page so you don't need to worry about sending your customer to another payment page or environment.

NOTE: PayPal Plus requires a vetting process, please contact PayPal to enable this product on your PayPal account, for sandbox this is not required.

Chapter 2. Before you begin

Before starting the technical integration you need to follow some steps to properly configure your testing environment using our Sandbox service.

PayPal Sandbox

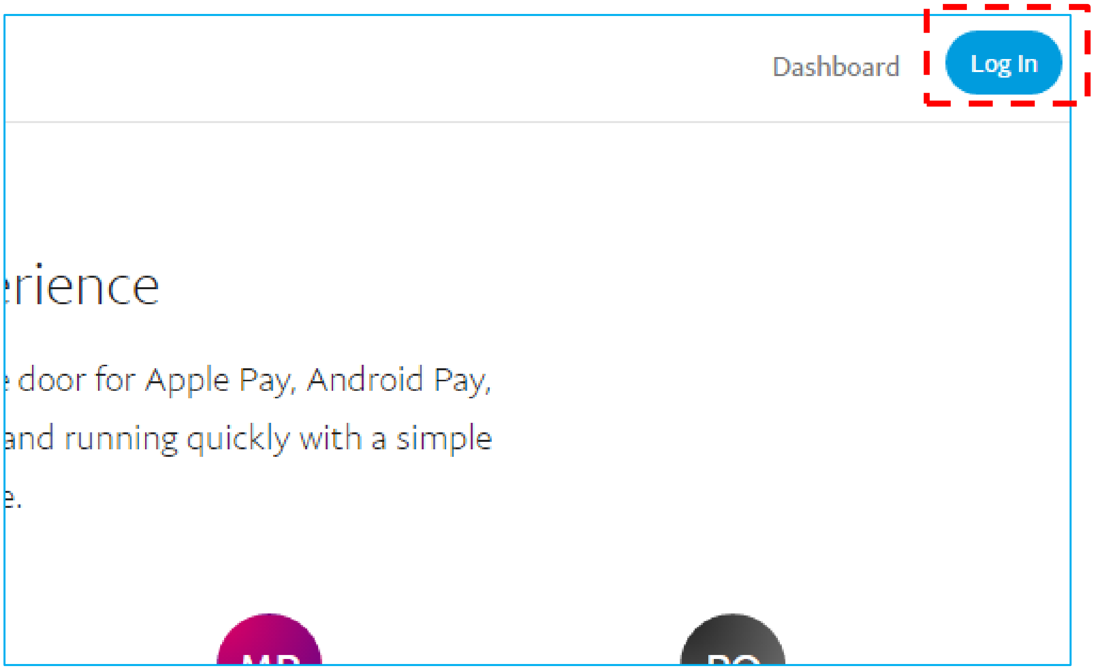
PayPal Sandbox is a testing environment where it mirrors the functionalities of production environment so you can fully test our products and API's without using real information or money.

You can integrate using Sandbox and stress the integration from begging to end, after everything is properly tested you just need to switch endpoints and API credentials to go live, you can read more about Sandbox environment here.

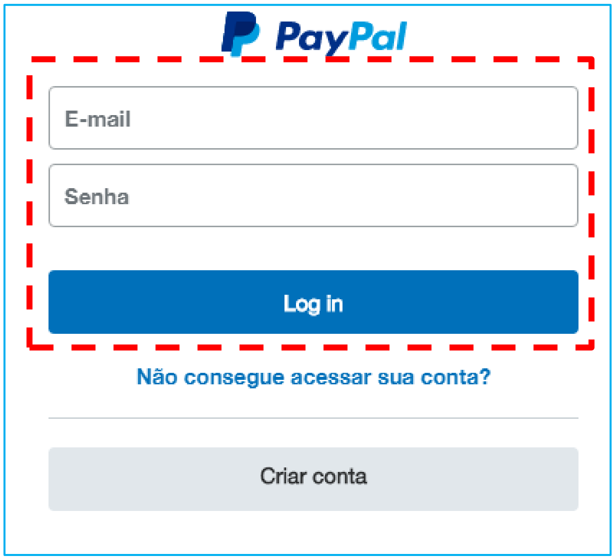
Creating a Merchant sandbox account

To test PayPal Plus solution you will need a Brazilian merchant Sandbox account, to create one go to https://developer.paypal.com/ and follow the steps bellow:

1. At the Developer Portal, click on the Log-in button if you are not already logged with your PayPal account:


Dashboard
Log In
rience
door for Apple Pay, Android Pay,
and running quickly with a simple
e.

2. Log-in with a real PayPal account:



PayPal

E-mail

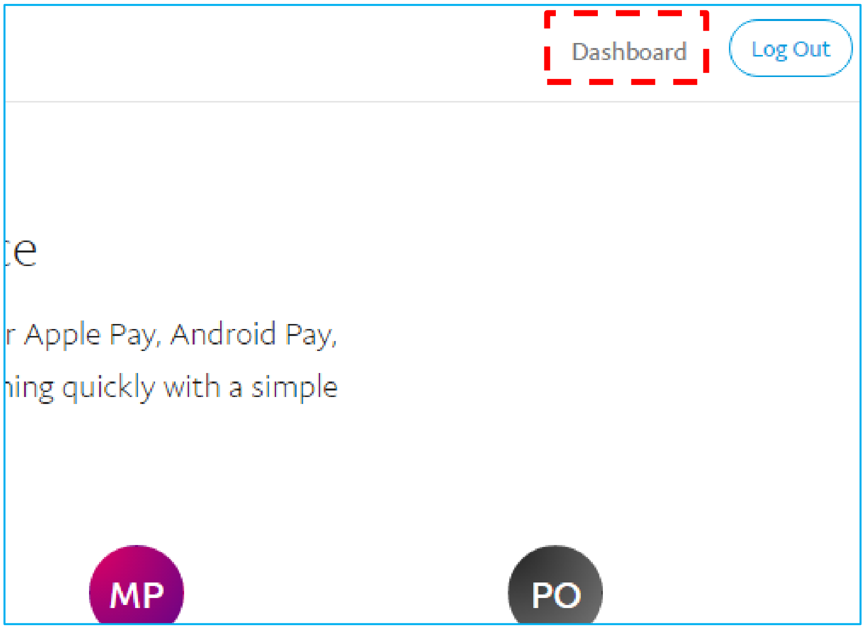
Senha

Log in

Não consegue acessar sua conta?

Criar conta

3. After logging, click on the "Dashboard" link.



Dashboard

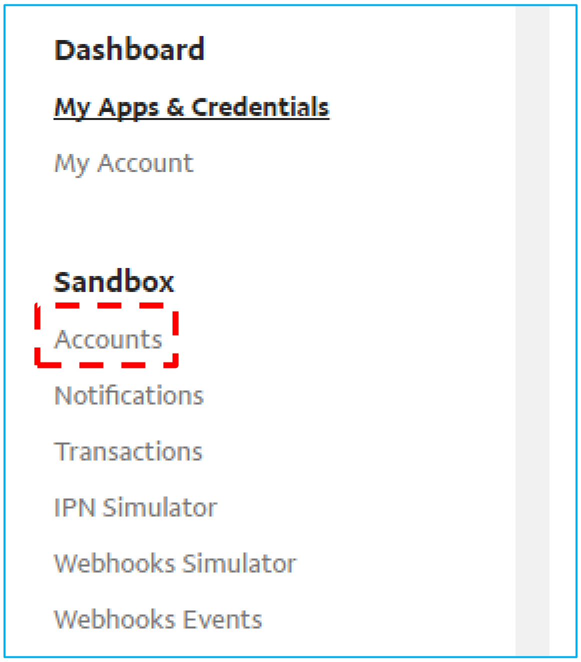
Log Out

:e

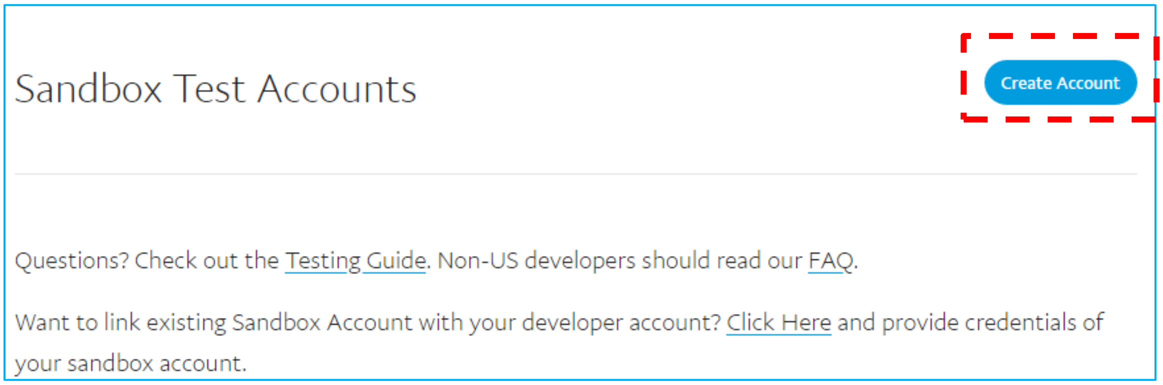
r Apple Pay, Android Pay,

ing quickly with a simple

4. Access the left-side menu and click on "Accounts" link under Sandbox label:



5. On the Sandbox Test Accounts page, click on the "Create Account" button:



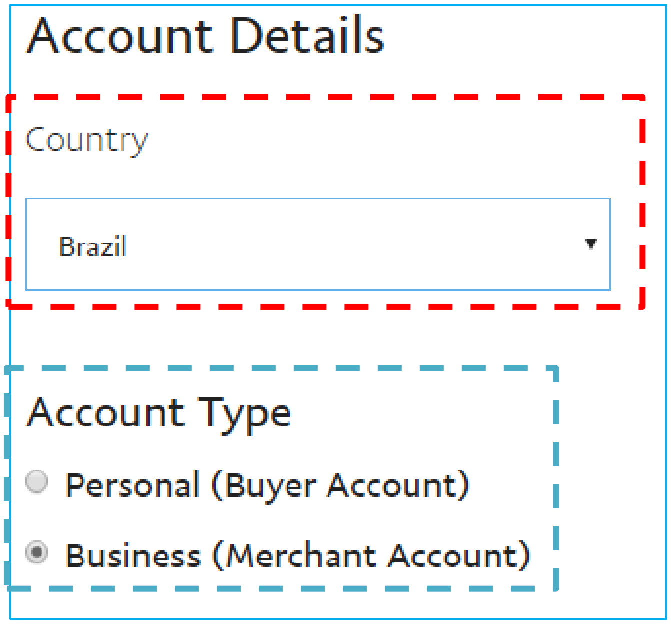
Sandbox Test Accounts

Questions? Check out the Testing Guide. Non-US developers should read our FAQ.

Want to link existing Sandbox Account with your developer account? Click Here and provide credentials of your sandbox account.

NOTE: Do not use the already created sandbox accounts, create a new one with the proper configuration of the next steps.

6. On the account creation page, make sure the field Country is selected as Brazil and the Account Type as Business, just like the image bellow (The other fields are generic and you can fill them as you please):



Account Details

Country

Brazil ▾

Account Type

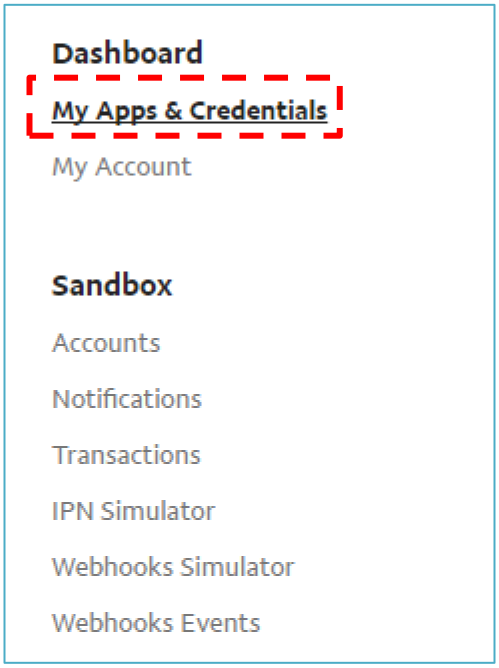
○ Personal (Buyer Account)

● Business (Merchant Account)

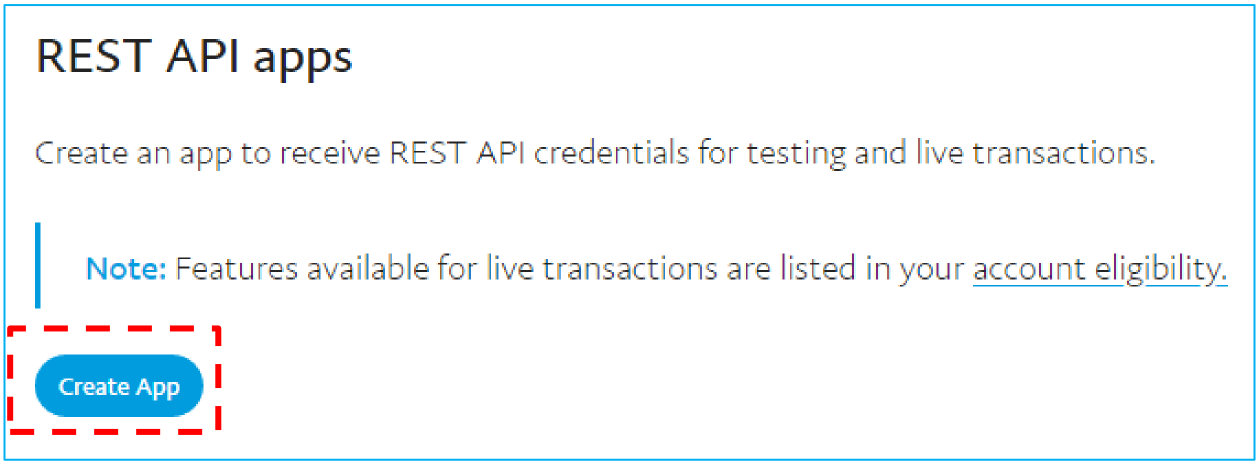
Generating API Credentials

Now you need to generate your Merchant API credentials to configure on your integration, continue on the developer portal and follow the steps bellow:

1. Check the left menu again and click on the “My Apps & Credentials” link:



2. On the Apps page, scroll down to REST API apps section and click on the “Create App” button:

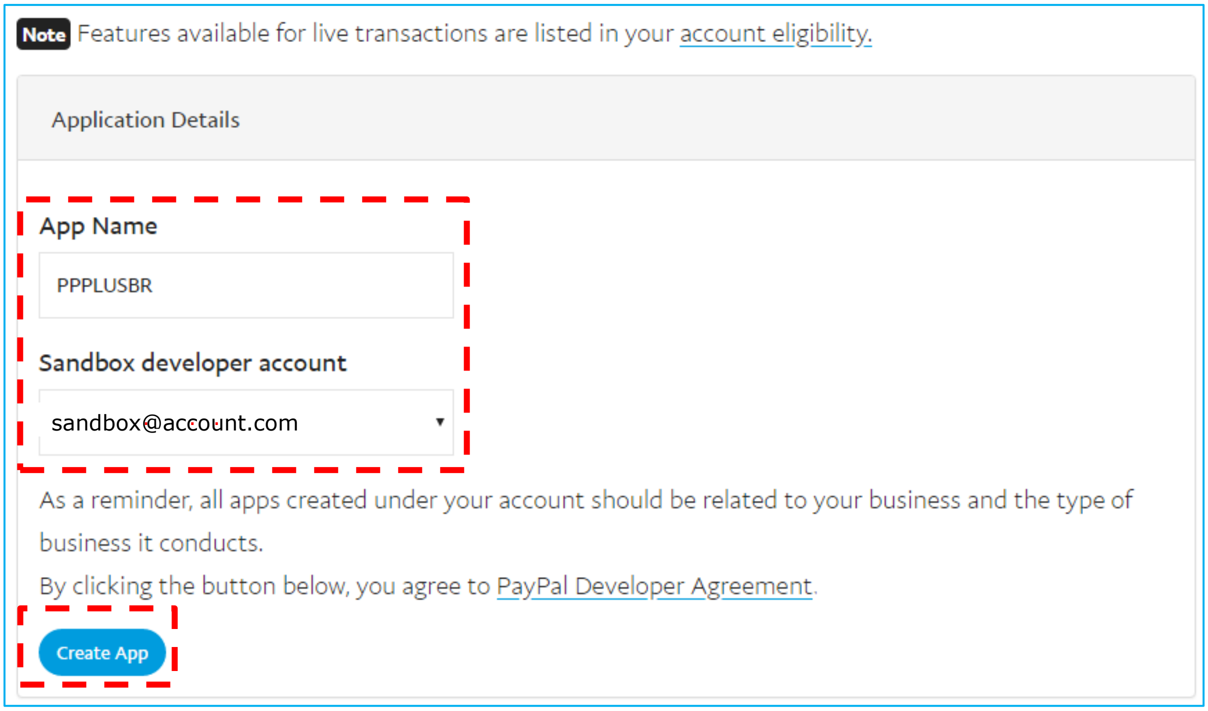


REST API apps

Create an app to receive REST API credentials for testing and live transactions.

Note: Features available for live transactions are listed in your account eligibility.

3. On the application details page, choose your desired name for the application and make sure you select your just created sandbox account on the Sandbox developer Account drop-down, after these steps click on the “Create App” button:



Note Features available for live transactions are listed in your account eligibility.

Application Details

App Name

PPPLUSBR

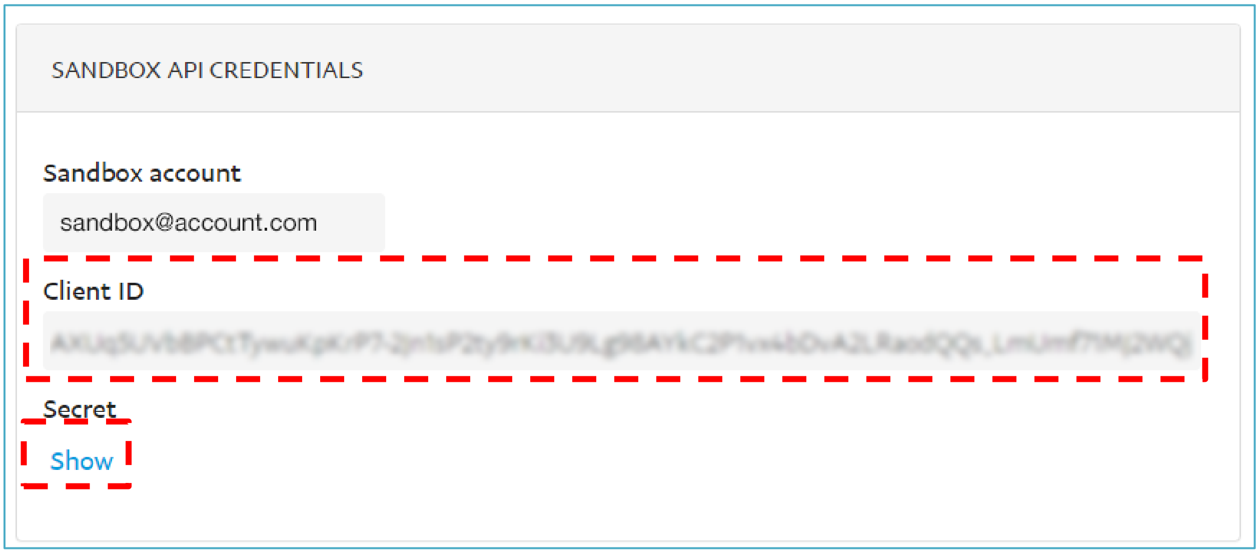
Sandbox developer account

sandbox@account.com ▾

As a reminder, all apps created under your account should be related to your business and the type of business it conducts.

By clicking the button below, you agree to PayPal Developer Agreement.

4. On the API Credentials page you will have access to your REST API Credentials (Client ID and Secret ID), copy your Client ID to a safe place then click on the "Show" button to have access to your Secret ID:



SANDBOX API CREDENTIALS

Sandbox account

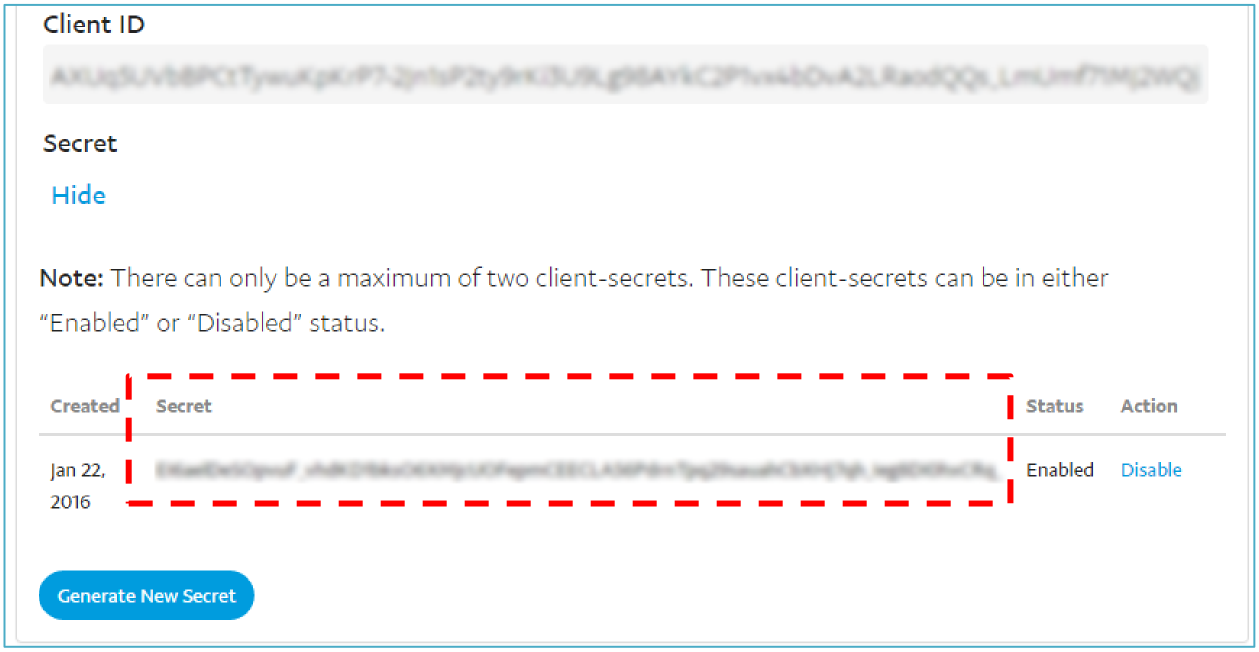
sandbox@account.com

Client ID

[redacted]

Secret

5. After clicking on the “Show” button, your Secret ID will appear under the button:



Client ID

[redacted]

Secret

Note: There can only be a maximum of two client-secrets. These client-secrets can be in either “Enabled” or “Disabled” status.

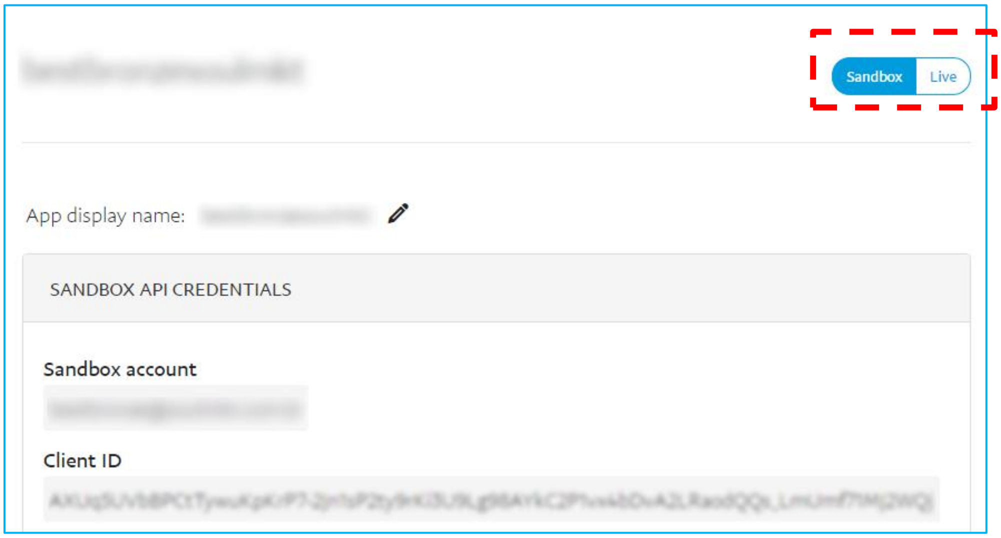
Created	Secret	Status	Action
Jan 22, 2016	[redacted]	Enabled	Disable

6. Save your Client ID and Secret ID on a safe place, you will use them to configure your integration.

Environments & API Endpoints

To switch between Sandbox or Live environment, you will need to change API credentials and API endpoints:

1. To generate Live API credentials just go to step 4 from the previous chapter and click no the Live button:



[redacted]

App display name: [redacted]

SANDBOX API CREDENTIALS

Sandbox account

[redacted]

Client ID

[redacted]

NOTE: If your merchant PayPal account was not the account used to generate sandbox credentials you will need to create a new sandbox account from the merchant PayPal account and generate your Live API credentials.

2. To switch between API endpoints for sandbox and live environment, just use the ones bellow:

Environment	**URI**
Sandbox	https://api.sandbox.paypal.com
Live	https://api.paypal.com

Chapter 3. Building a payment solution with PayPal Plus

The following sections describe how to develop a payment solution with PayPal Plus as follows:

- OVERVIEW – Lists each task to complete and basic information about each task.
- PAYMENT FLOW – Illustrates the payment flow that you develop for a PayPal Plus integration.
- INTEGRATING PAYPAL PLUS – Each sub-section provides detailed information about each task to complete to build your payment flow and integrate with PayPal Plus.

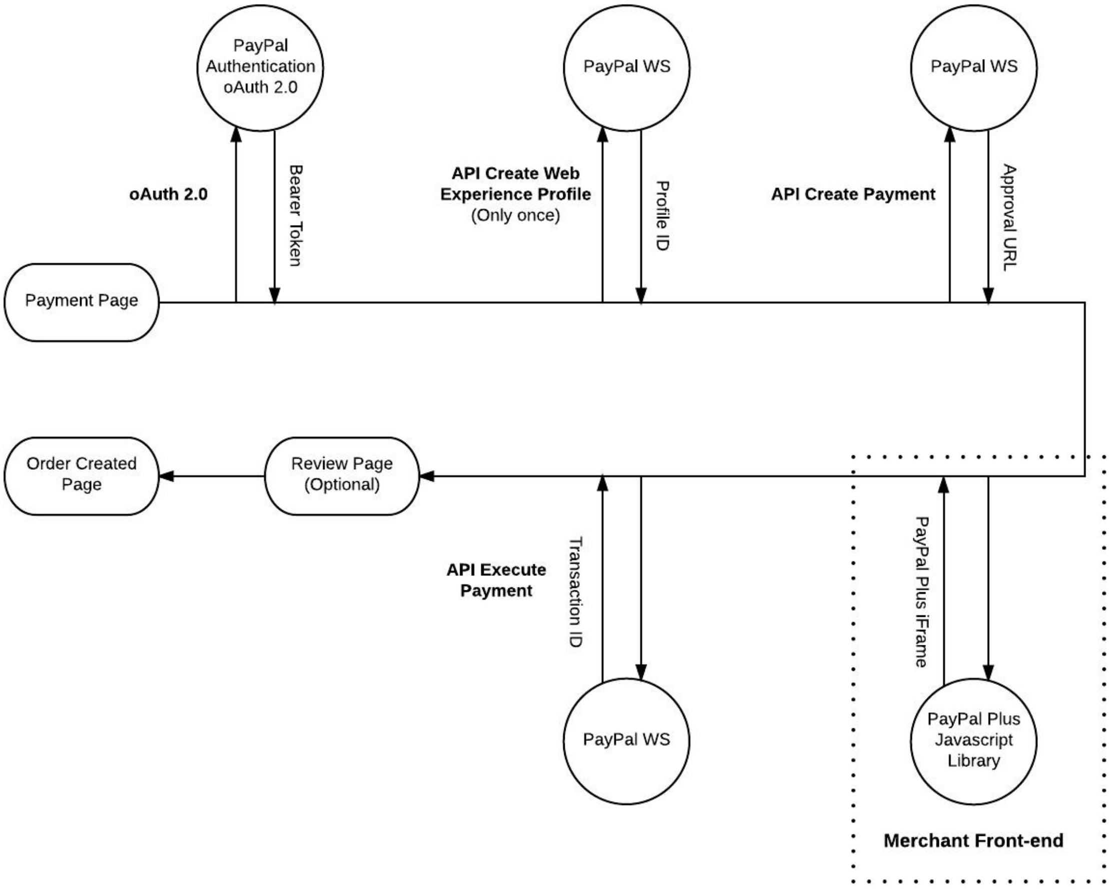
Overview

To use the PayPal Plus hosted payment selection page in your payment processing flow, you need to plan to complete the following tasks:

Task to complete	Description	For detailed instructions, see...
Get an access token	Initiate the payment process by issuing a request for an access token	Requesting an access token
Issue a create payment request	Initial payment must contain amount, currency, items, and shipping address (if applicable)	Creating a payment
Embed JavaScript code from the PayPal Plus JS library in your checkout page	Provide the initialization parameters needed to properly load and render the payment selection page iFrame. The payment selection page is rendered on demand and based on the approval URL returned by the create payment request.	Integrating the Payment Selection page
Create a Continue button	Create a button outside the payment selection page to implement the DoContinue function. (PayPal recommends deactivating this button until all payment information is filled out by the buyer in the payment selection page.)	Integrating the external Continue button
Create a Javascript listener	Implement a Javascript listener to retrieve all the iFrame interaction on the client-side.	Integrating the Javascript listener
Complete the transaction	Complete the transaction by issuing a request to execute the payment.	Executing the payment

Payment flow with PayPal Plus

The following diagram illustrates the overall payment flow to correspond with the tasks outlined in the preceding Overview section:



Integrating with PayPal Plus

The following sections provide detailed information about each task to complete to build your payment flow and integrate with PayPal Plus.

Requesting an access token

As described in Make your first call on the PayPal Developer portal, you must have a valid access token to make API requests. You can use your application’s `client_id` and `secret` to request an access token as shown in the following sample request. In the request body, set `grant_type` to `client_credentials`.

EXAMPLE ACCESS TOKEN REQUEST

```
ENDPOINT: https://api.sandbox.paypal.com/v1/oauth2/token \
 -H "Accept: application/json" \
 -u "<Client-Id>:<Secret-Id>" \
 -d "grant_type=client_credentials"
```

NOTE: For partner's integrations, you need to specify a unique code provided by PayPal at the header of the request on the "PayPal-Partner-Attribution-Id" parameter like the example bellow:

EXAMPLE ACCESS TOKEN REQUEST FOR PARTNERS

```
ENDPOINT: https://api.sandbox.paypal.com/v1/oauth2/token \
 -H "Accept: application/json" \
 -H "PayPal-Partner-Attribution-Id: <Your-BN-Code>" \
 -u "<Client-Id>:<Secret-Id>" \
 -d "grant_type=client_credentials"
```

EXAMPLE ACCESS TOKEN RESPONSE

```
{
 "scope": "https://api.paypal.com/v1/payments/.*
    https://api.paypal.com/v1/vault/credit-card
    https://api.paypal.com/v1/vault/credit-card/.*",
 "access_token": "<Access-Token>",
 "token_type": "Bearer",
 "expires_in": 32400
}
```

The returned access token is valid for the number of seconds specified in the `expires_in` response value. You must have a valid access token to make API requests. Request a new token when the current one expires.

For more information, refer to the following resources in the REST API Reference on the PayPal Developer Portal:

API Reference	Description	URL
REST API Authentication and headers	Optional and mandatory parameters to set on the OAuth 2.0 authentication	RESTP API Authentication and headers
Make your first call	Real example of OAuth 2.0 authentication	Make your first call

Creating a payment

After receiving your access token, you can use it to issue a request to create a payment. When you issue the create payment request, be sure to set the following fields to the values shown in the following sample: `intent`, `payment_method`, `currency`, `allowed_payment_method`, `return_url`, `cancel_url`.

Example Create Payment Request

```
ENDPOINT: https://api.sandbox.paypal.com/v1/payments/payment \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer <Access-Token>' \
-d '{
"intent":"sale",
   "payer":{
      "payment_method":"paypal"
   },
    "application_context": {
        "brand_name":"<<Store Name>>",
        "shipping_preference":"SET_PROVIDED_ADDRESS"
    },
    "transactions":[
      {
         "amount":{
            "currency":"BRL",
            "total":"30.00",
            "details":{
              "shipping":"10.00",
              "subtotal":"20.00”
            }
         },
         "description":"Order #942342 from storeURL",
         "payment_options":{
            "allowed_payment_method":"IMMEDIATE_PAY"
         },
         "invoice_number": "942342",
         "item_list":{
            "shipping_address": {
                "recipient_name" : "Thiago Gustavo Campos",
                "line1": "Avenida dos Tarumãs, 32 - apt 123",
                "line2": "Bairro",
                "city": " Sâo Paulo",
                "country_code": "BR",
                "postal_code": " 78556224",
                "state": "SP",
                "phone": "(66)9371-5868"
            },
            "items":[
               {
                  "name":"Product",
                  "description":"Product description",
                  "quantity":"2",
                  "price":"10.00",
                  "sku":"product_id_99",
                  "currency":"BRL"
               }
            ]
         }
      }
   ],
   "redirect_urls":{
      "return_url":"http://www.<<Store URL>>.com",
      "cancel_url":"http://www.<<Store URL>>.com"
   }
}
```

NOTE 01: Pay attention to the parameters "payment_method" "payment_options" and the "redirect_urls", they are all mandatory. The URL's inside the "redirect_urls" will need to contain your store URL.

NOTE 02: If you sell digital goods and do not provide a shipping address set the "shipping_preference" parameter to "NO_SHIPPING".

EXAMPLE CREATE PAYMENT RESPONSE

```
{
    "id": "PAY-3A3234483P2338009KTTFX7Q",
    "create_time": "2015-02-19T21:56:14Z",
    "update_time": "2015-02-19T21:56:14Z",
    "state": "created",
    "intent": "sale",
    "payer": {
        "payment_method": "paypal",
        "payer_info": {
            "shipping_address": {}
        }
    },
    "transactions": [
        {
            "amount": {
                "total": "95.00",
                "currency": "BRL",
                "details": {
                    "subtotal": "75.00",
                    "shipping": "11.00"
                }
            },
           "description": "This is the payment transaction description",
           "item_list": {
                "items": [
                    {
                        "name": "handbag",
                        "sku": "product34",
                        "price": "75.00",
                        "currency": "BRL",
                        "quantity": "1",
                        "description": "red diamond"                     }
                ],
                "shipping_address": {
                    "recipient_name": "Thiago Gustavo Campos",
                    "line1": "Avenida dos Tarumãs, 32",
                    "line2": "apt 42",
                    "city": "Sinop",
                    "state": "São Paulo",
                    "phone": "(66)9371-5868",
                    "postal_code": "78556-224",
                    "country_code": "BR"
                }
            },
            "related_resources": []
        }
    ],
    "links": [
        {
            "href": "https://api.sandbox.paypal.com/v1/payments/payment/PAY-
3A3234483P2338009KTTFX7Q",
            "rel": "self",
            "method": "GET"
        },
        {
            "href": "https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_express-
checkout&token=EC-82237386YH588524U",
            "rel": "approval_url",
            "method": "REDIRECT"
        },
        {
            "href": "https://api.sandbox.paypal.com/v1/payments/payment/PAY-
3A3234483P2338009KTTFX7Q/execute",
            "rel": "execute",
            "method": "POST"
        }
    ]
}
```

On the createPayment API response there will be 3 URL's on the HATEOAS format, pay attention to those URL's as they are important for the next steps of the flow, bellow follows a detailed explanation of the use:

URL (links[rel])	Action
self	(Optional) Used to obtain details about a payment session.
approval_url	Used on the PayPal Plus iFrame creation so store it to use it later
execute	Used to execute the payment, the URL will be the endpoint of the API so store it as well

Integrating the Payment Selection page

The payment selection page is the actual payment iFrame where costumers enter their credit card information. To render the iFrame, you must complete the following tasks:

Paypal Plus Javascript library

Include the JavaScript library on your checkout page using the following script:

```
<script
src="https://www.paypalobjects.com/webstatic/ppplusdcc/ppplusdcc.min.js"
type="text/javascript">

</script>
```

When you load the `ppplusdcc.min.js` JavaScript library, it creates a new object called `PAYPAL.apps.PPP`. The `PPP` object defines the API described in the following section.

HTML DIV Placeholder

Create a HTML DIV on your checkout page where the iFrame will be embedded:

```
<div id="ppplusDiv"> </div>
```

Paypal Plus Javascript object

Render the PayPal Plus iFrame using the following Javascript example. Include the `approval_url` returned from the createPayment API previously explained.

SAMPLE CLIENT-SIDE JAVASCRIPT TO OPEN IFRAME

```
<script type="application/javascript">
  var ppp = PAYPAL.apps.PPP({
    "approvalUrl": "https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_express-
checkout&token=EC-82237386YH588524U",
    "placeholder": "ppplusDiv",
    "mode": "sandbox",
     "payerFirstName": "customerFirstName",
     "payerLastName": " customerLastName",
     "payerPhone": " customerPhone",
     "payerTaxId": " customerTaxId",
     "payerTaxIdType": "BR_CPF",
     "language": "pt_BR",
     "country": "BR",
     "rememberedCards": "customerRememberedCardHash",
     …etc
     });
</script>
```

NOTE: These are some example parameters, there are many others, the section bellow will explain every one of them.

Configure the Continue button as described in Integrating the external Continue button on page 23.

The components used in the preceding steps to include the Payment Selection iFrame in your checkout page (`ppplusdcc.min.js` JavaScript library and `PAYPAL.apps.PPP` API) are described in the following sections.

PayPal Plus API

The PayPal Plus API requires a configuration object that must include required parameters and can also include optional parameters as described in the following table:

Parameter	**Type**	**Description**
`approvalUrl`	string	**(Required)** The `approval_url` returned by the create payment call. See Creating a payment on page 14 for a sample response that includes the approval_url.
`placeholder`	string	**(Required)** The ID of a DOM object where the payment selection page will be embedded in your checkout page. The payment selection page adapts to the width of the specified object.
`payerFirstName`	string	**(Required)** First name of the payer providing payment information for checkout. This name must match the name on the credit card.
`payerLastName`	string	**(Required)** Last name of the payer providing payment information for checkout. This name must match the name on the credit card.
`payerEmail`	string	**(Required)** Email address of the payer providing payment information for checkout.
`payerPhone`	string	**(Required)** Phone number of the payer providing payment information for checkout. Required for seller protection.

Parameter	Type	Description
payerTaxId	string	**(Required)** Tax identification number of the payer providing payment information for checkout. Required for Brazilian citizens, leave blank for non-citizens.
payerTaxIdType	string	**(Required)** Type of Tax ID. Allowed values are: BR_CPF or BR_CNPJ. Required for Brazilian citizens, leave blank for non-citizens.
language	string	**(Required)** Language in which to render the PayPal Plus payment selection page. This should be the language used in your website and checkout page so that all content is consistent for the buyer. Allowed values are: en_US, pt_BR, de_DE (defaults to en_US).
country	string	**(Required)** Country in which the PayPal Plus service will be deployed. Allowed values: US, BR, DE (defaults to US).
mode	string	**(Required)** Depending on the mode, the library will load the payment selection page from different locations. Allowed values are live and sandbox. If you specify live, it is loaded from paypal.com whereas sandbox loads it from sandbox.paypal.com. The library also issues a warning to the console if the mode is set to sandbox. In live mode, warnings are issued only if required parameters are missing.
onContinue	callback or string	If a string is provided, that string will be interpreted as the ID of a form in the document which will be submitted when the inside Continue button is clicked. If a callback is provided, that callback will be invoked when the inside Continue button is clicked.
enableContinue	callback or string	**(Required)** If a string is provided, that string will be interpreted as the ID of the outside Continue button (may be any element) which will be enabled when a payment selection is made. If a callback is provided, that callback will be invoked when the outside Continue button is clicked.
disableContinue	callback or string	**(Required)** If a string is provided, that string will be interpreted as the ID of the outside Continue button (may be any element) which will be disabled if no payment is selected. If a callback is provided, that callback will be invoked then.

Parameter	Type	Description
`rememberedCards`	string	Holds the IDs needed for PayPal to retrieve the user’s remembered card data. Always send the latest ID you have stored on your database.
`merchantInstallmentSelectionOptional`	boolean	Let the costumer change the number of installments selected, this parameter is used with the “merchantInstallmentSelection” parameter.
`merchantInstallmentSelection`	integer	Set a predefined number of installments to the customer, this parameter is used with the “merchantInstallmentSelectionOptional” parameter.
`iframeHeight`	string	This will limit the iFrame height to a specific value, if the iFrame content overflow the limit, a scroll bar will be created. Consider the height in pixels and set it like a string with “” (Limit 300 pixels).

External Continue Button

In the script to render the payment selection page, be sure to set the following parameters to these values:

```
disableContinue: "continueButton",
enableContinue: "continueButton",
```

When the external Continue button is pressed, a response object that contains the payer’s ID, the rememberedCards token, and the installment selection is posted to the checkout page if execution of payment can proceed. If payment execution cannot proceed, an error is displayed.

Add the following script to your external Continue button:

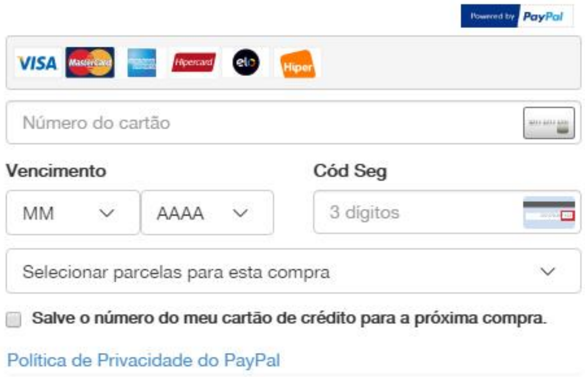
```
<button>
      type="submit"
      id="continueButton"
      onclick="ppp.doContinue(); return false;"> Checkout
</button>
```

The ppp.doContinue() function will trigger the iFrame validation.

Sample Payment Selection Pages

When integration and configuration are complete, the following images show examples of how the Payment Selection iFrame may look when it is included in your checkout page:

New Card



Número do cartão

Vencimento

MM

AAAA

Cód Seg

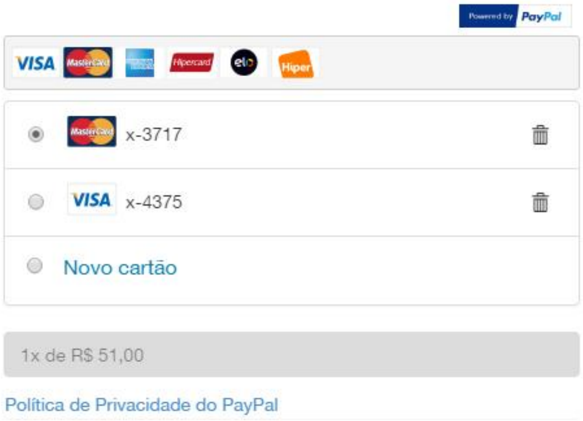
3 dígitos

Selecionar parcelas para esta compra

□ Salve o número do meu cartão de crédito para a próxima compra.

Política de Privacidade do PayPal

Saved Card



● x-3717

○ VISA x-4375

○ Novo cartão

1x de R$ 51,00

Política de Privacidade do PayPal

Integrating the Javascript iFrame listener

For PayPal Plus iFrame interaction, you will need to implement a client-side Javascript listener that will receive every postMessage() from PayPal Plus application.

The following sample code examples how to handle the response object that returns when the external Continue button is pressed, also includes how to handle iFrame errors:

Sample Client-Side JavaScript to Handle Post Message

```
<script>

if (window.addEventListener) {

    window.addEventListener("message", messageListener, false);

    console.log("addEventListener successful", "debug");

} else if (window.attachEvent) {

    window.attachEvent("onmessage", messageListener);

    console.log("attachEvent successful", "debug");

} else {

    console.log("Could not attach message listener", "debug");

    throw new Error("Can't attach message listener");

}

function receiveMessage(event)
{

    try {

        var message = JSON.parse(event.data);

        if (typeof message['cause'] !== 'undefined') { //iFrame error handling

            ppplusError = message['cause'].replace (/['"]+/g,""); //log & attach this error into the order
if possible

     // <<Insert Code Here>>

            switch (ppplusError)

                {

                    case "INTERNAL_SERVICE_ERROR": //javascript fallthrough
                    case "SOCKET_HANG_UP": //javascript fallthrough
                    case "socket hang up": //javascript fallthrough
                    case "connect ECONNREFUSED": //javascript fallthrough
                    case "connect ETIMEDOUT": //javascript fallthrough
                    case "UNKNOWN_INTERNAL_ERROR": //javascript fallthrough
                    case "fiWalletLifecycle_unknown_error": //javascript fallthrough
                    case "Failed to decrypt term info": //javascript fallthrough
                    case "RESOURCE_NOT_FOUND": //javascript fallthrough
```

```
          case "INTERNAL_SERVER_ERROR":
          alert ("Ocorreu um erro inesperado, por favor tente novamente."); //pt_BR
          //Generic error, inform the customer to try again; generate a new approval_url and
reload the iFrame.
          // <<Insert Code Here>>
          break;

          case "RISK_N_DECLINE": //javascript fallthrough
          case "NO_VALID_FUNDING_SOURCE_OR_RISK_REFUSED": //javascript fallthrough
          case "TRY_ANOTHER_CARD": //javascript fallthrough
          case "NO_VALID_FUNDING_INSTRUMENT":
          alert ("Seu pagamento não foi aprovado. Por favor utilize outro cartão, caso o
problema persista entre em contato com o PayPal (0800-047-4482)."); //pt_BR
          //Risk denial, inform the customer to try again; generate a new approval_url and
reload the iFrame.
          // <<Insert Code Here>>
          break;

          case "CARD_ATTEMPT_INVALID":
          alert ("Ocorreu um erro inesperado, por favor tente novamente."); //pt_BR
          //03 maximum payment attempts with error, inform the customer to try again; generate a
new approval_url and reload the iFrame.
          // <<Insert Code Here>>
          break;

          case "INVALID_OR_EXPIRED_TOKEN":
          alert ("A sua sessão expirou, por favor tente novamente."); //pt_BR
          //User session is expired, inform the customer to try again; generate a new
approval_url and reload the iFrame.
          // <<Insert Code Here>>
          break;

          case "CHECK_ENTRY":
          alert ("Por favor revise os dados de Cartão de Crédito inseridos."); //pt_BR
          //Missing or invalid credit card information, inform your customer to check the
inputs.
          // <<Insert Code Here>>
          break;

          default: //unknown error & reload payment flow
          alert ("Ocorreu um erro inesperado, por favor tente novamente."); //pt_BR
          //Generic error, inform the customer to try again; generate a new approval_url and
reload the iFrame.
          // <<Insert Code Here>>

        }

    }

    if (message['action'] == 'checkout') { //PPPlus session approved, do logic here

    var rememberedCard = null;
      var payerID = null;
      var installmentsValue= null;

      rememberedCard = message['result']['rememberedCards']; //save on user BD record
      payerID = message['result']['payer']['payer_info']['payer_id']; //use it on executePayment API

      if("term" in message){
        installmentsValue = message['result']['term']['term']; //installments value
```

```
            } else {
                installmentsValue=1; //no installments
            }

            /* Next steps:

                1) Save the rememberedCard value on the user record on your Database.
                2) Save the installmentsValue value into the order (Optional).
                3) Call executePayment API using payerID value to capture the payment.

            */

            // <<Insert Code Here>>

        }

    } catch (e){ //treat exceptions here

     // <<Insert Code Here>>

    }

}

</script>
```

If the information inputted by the customer is valid and there aren't errors the iFrame library will return a JSON like the one below, you will use the listener above to capture it:

SAMPLE JSON RETURNED BY PAYPAL PLUS IFRAME

```
{
   "action":"checkout",
   "result":{
      "id":"EC-2KC16623VX945640J",
      "state":"APPROVED",
      "cart_id":"EC-2KC16623VX945640J",
      "shipping_address":{
         "recipient_name":"Arthur Puhlmann",
         "id":"3976576581",
         "line1":"Rua teste 1234",
         "city":"São Paulo",
         "state":"SP",
         "postal_code":"12312312",
         "country_code":"BR",
         "normalization_status":"UNKNOWN",
         "default_address":true,
         "preferred_address":false,
         "disable_for_transaction":false
      },
      "payer":{
         "payment_method":"paypal",
         "status":"UNVERIFIED",
         "payer_info":{
            "email":"apuhlmann@paypal.com",
            "first_name":"Arthur",
            "last_name":"Puhlmann",
            "payer_id":"C8PPNG2M73WTC",
            "shipping_address":{
               "recipient_name":"Arthur Puhlmann",
               "id":"3976576581",
               "line1":" Rua teste 1234",
               "city":"São Paulo",
               "state":"SP",
               "postal_code":"12312312",
               "country_code":"BR",
               "normalization_status":"UNKNOWN",
               "default_address":true,
               "preferred_address":false,
               "disable_for_transaction":false
            },
            "phone":"1111111111",
            "tax_id_type":"BR_CPF",
            "tax_id":"22954292377",
            "country_code":"BR"
         },
         "funding_option":{
            "id":"115b2091dd3947b3865a8050ec255556",
            "funding_sources":[
               {
                  "funding_mode":"INSTANT_TRANSFER",
                  "funding_instrument_type":"PAYMENT_CARD",
                  "amount":{
                     "value":"52.00",
                     "currency":"BRL"
                  },
                  "soft_descriptor":"MERCHANTBRA"
               }
            ]
         }
      },
      "payment_approved":true,
      "declined_instruments":[
         {
            "funding_instrument_type":"CREDIT",
            "decline_type":"PAYPAL_ELIGIBILITY",
            "reason_code":"PPCREDIT_NOT_IN_WALLET_NOT_ELIGIBLE"
         }
      ],
      "rememberedCards":"vNSA9QzsuHXEIk_zcxT8zM0w3VKuH9vKdy55_aIfRto3-9ClFdFyrXg8TJyTXQv8_o0Y2OpI-
Jmh0EZndqVRLXAEWjipQEmjikrynJd2FK4ojYea",
      "term":{
         "term":2,
         "monthly_payment":{
            "value":"26.00",
            "currency":"BRL"
         }
      }
   }
}
```

There are 3 important parameters on this JSON, they are:

Parameter	**Action**
payer_id (Mandatory)	In this parameter will be the key to execute the payment, call the API executePayment using this value to conclude the payment.
rememberedCards (Optional)	This parameter will contain a hash, it represents the saved cards from the costumer on your store, always save it on your database attached with customer record and pass the value on the iFrame creation.
term (Optional)	This parameter will return the installments option chosen by the customer, you will use it to know in how many times he will be paying this purchase, save the value on the order.If you don't accept installments or his credit card doesn't this parameter will not be returned.

Handling errors on the iFrame

The PayPal Plus iFrame can also return some errors like invalid credit card or data, risk declines or general errors, the errors will be returned on a JSON like the one below:

SAMPLE JSON RETURNED BY PAYPAL PLUS IFRAME IN CASE OF ERROR

```
{
   "action":"onError",
   "cause":"\"TRY_ANOTHER_CARD\”"
}
```

> **NOTE:** Pay attention as the cause parameter will always return the value between a \"value\” string, don’t forget to consider the slash and the quotation mark.

Use the “cause” parameter to filter the real cause of the error and give the proper action, the most common errors are:

Cause	**Action**
NO_VALID_FUNDING_INSTRUMENT RISK_N_DECLINE TRY_ANOTHER_CARD NO_VALID_FUNDING_INSTRUMENT	PayPal risk decline, finish the order with payment not approved or offer PayPal Express Checkout as a payment method.
CHECK_PROVIDED_USER_DATA INVALID_DATA_FIRST_NAME	Verify integration, a mandatory parameter is missing.
PPPLUS_NOT_AVAILABLE_FOR_MERCHAN	Product not enabled, contact your PayPal account manager.
SET_INSTALLMENTS_FAIL	Verify integration, installments parameter doesn’t match currently installments plan.
fiWalletLifecycle_unknown_error	Call the javascript doContinue function again, if the error persists, check the integration.
PAYMENT_ALREADY_DONE	The payment for this session is already done, check the integration.
CHECK_ENTRY	Customer forgot a credit card field or mistyped.

Executing the payment

To complete the payment flow, you should issue a request to execute the payment API. Payment execution can be initiated from the final Order Review page or in the background upon return from the hosted pages (when there is no Order Review page). When you execute the payment, any updates to the shipping address and the shipping and handling charges should be submitted with the request. There must be no changes to the item list.

The `payer_id` value needed to execute the payment is provided in the response from the JSON returned by the iFrame Javascript. The following sample shows how to issue a request to execute payment.

Example Execute Payment Request

```
ENDPOINT: https://api.sandbox.paypal.com/v1/payments/payment/<Payment-Id>/execute/ \
-H "Content-Type:application/json" \
-H "Authorization: Bearer <Access-Token>" \
-d '{"payer_id" : "<Payer-Id>" }'
```

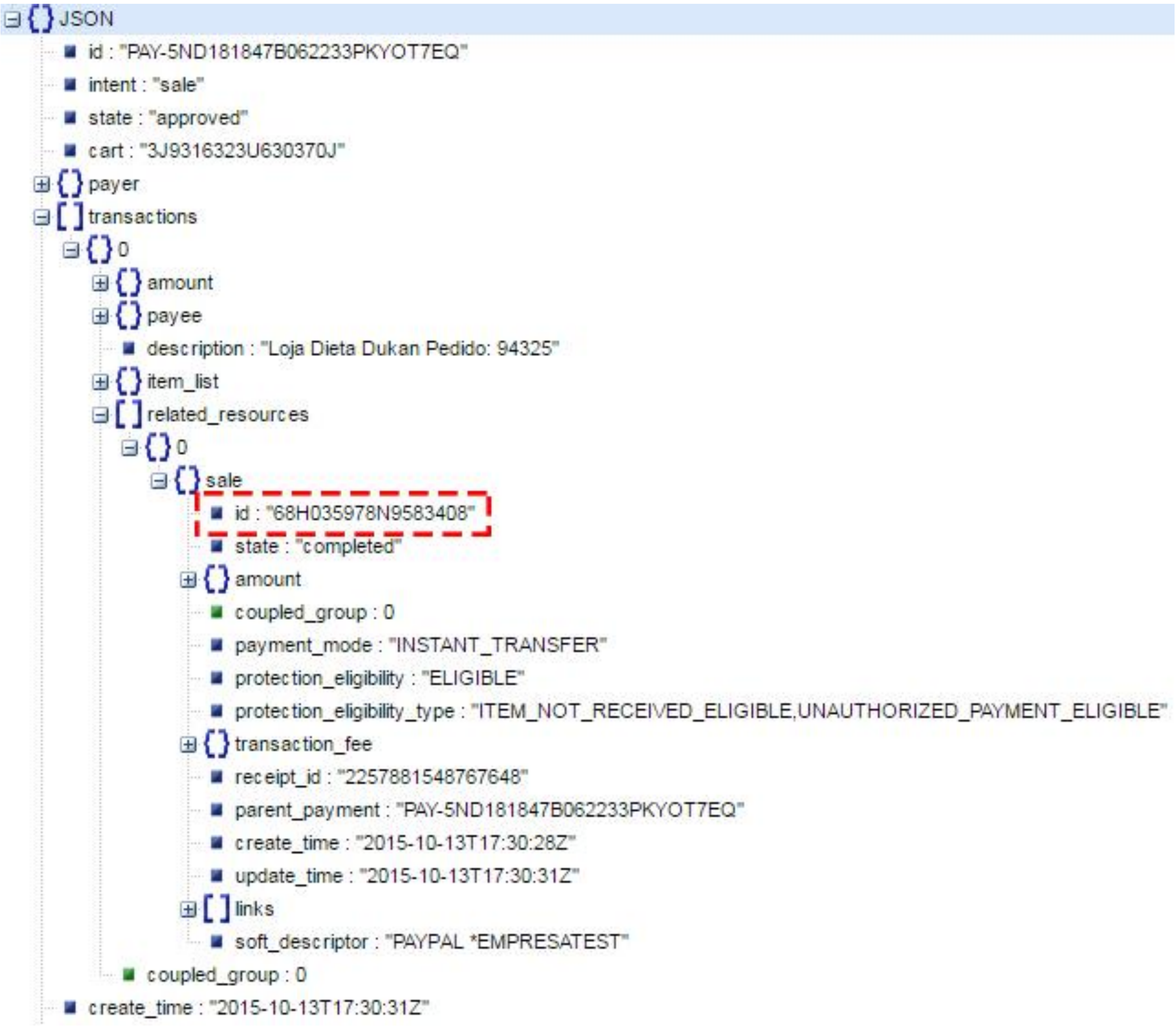
Example Execute Payment Response

```
{
  "id": "PAY-7V4081612V751425TKS7OPEQ",
  "create_time": "2015-01-20T23:41:06Z",
  "update_time": "2015-01-20T23:43:58Z",
  "state": "approved",
  "intent": "sale",
  "payer": {
    "payment_method": "paypal",
    "payer_info": {
      "email": "test123456781111@paypal.com",
      "first_name": "Max",
      "last_name": "Santos",
      "payer_id": "SUYCECXSW5XMA",
      "shipping_address": {
        "line1": "Gregório Rolim de Oliveira, 42",
        "line2": "JD Serrano II",
        "city": "Votorantim",
        "state": "São Paulo",
        "postal_code": "18117-134",
        "country_code": "BR",
        "recipient_name": "Max Santos"
      }
    }
  },
  "transactions": [
    {
      "amount": {
        "total": "95.00",
        "currency": "BRL",
        "details": {
          "subtotal": "75.00",
          "tax": "6.00",
          "shipping": "11.00",
          "handling_fee": "1.00",
          "insurance": "1.00",
          "shipping_discount": "1.00"
        }
      },
      "description": "This is the payment transaction description",
      "item_list": {
        "items": [
          {
            "name": "handbag",
            "sku": "product34",
            "price": "75.00",
            "currency": "BRL",
            "quantity": "1",
            "description": "red diamond",
            "tax": "6.00"
          }
        ],
        "shipping_address": {
          "recipient_name": "Max Santos",
          "line1": "Gregório Rolim de Oliveira, 42",
          "line2": "JD Serrano II",
          "city": "Votorantim",
          "state": "São Paulo",
          "postal_code": "18117-134",
          "country_code": "BR"
        }
      },
      "related_resources": [
        {
          "sale": {
            "id": "2KV752851V792590K",
            "create_time": "2015-01-20T23:41:07Z",
            "update_time": "2015-01-20T23:43:58Z",
            "amount": {
              "total": "95.00",
              "currency": "BRL"
            },
            "payment_mode": "INSTANT_TRANSFER",
            "state": "completed",
            "protection_eligibility": "ELIGIBLE",
            "protection_eligibility_type":
"ITEM_NOT_RECEIVED_ELIGIBLE,UNAUTHORIZED_PAYMENT_ELIGIBLE",
            "parent_payment": "PAY-7V4081612V751425TKS7OPEQ",
            "links": [
              {
                "href": "https://api.sandbox.paypal.com/v1/payments/sale/2KV752851V792590K",
                "rel": "self",
                "method": "GET"
              },
              {
                "href":
"https://api.sandbox.paypal.com/v1/payments/sale/2KV752851V792590K/refund",
                "rel": "refund",
                "method": "POST"
              },
              {
                "href": "https://api.sandbox.paypal.com/v1/payments/payment/PAY-
7V4081612V751425TKS7OPEQ",
                "rel": "parent_payment",
                "method": "GET"
              }
            ]
          }
        }
      ]
    }
```

```
  ],
  "links": [
    {
      "href": "https://api.sandbox.paypal.com/v1/payments/payment/PAY-7V4081612V751425TKS7OPEQ",
      "rel": "self",
      "method": "GET"
    }
  ]
}
```

Execute payment API response

On its response, pay attention to the "id" parameter highlighted on the image below, it represents the unique transaction ID of this order, always save it on the order detail so you can correlate with PayPal admin area and ask for further assistance if needed.



```
⊟ { } JSON
    ■ id : "PAY-5ND181847B062233PKYOT7EQ"
    ■ intent : "sale"
    ■ state : "approved"
    ■ cart : "3J9316323U630370J"
  ⊞ { } payer
  ⊟ [ ] transactions
    ⊟ { } 0
      ⊞ { } amount
      ⊞ { } payee
        ■ description : "Loja Dieta Dukan Pedido: 94325"
      ⊞ { } item_list
      ⊟ [ ] related_resources
        ⊟ { } 0
          ⊟ { } sale
              ■ id : "68H035978N9583408"
              ■ state : "completed"
            ⊞ { } amount
              ■ coupled_group : 0
              ■ payment_mode : "INSTANT_TRANSFER"
              ■ protection_eligibility : "ELIGIBLE"
              ■ protection_eligibility_type : "ITEM_NOT_RECEIVED_ELIGIBLE,UNAUTHORIZED_PAYMENT_ELIGIBLE"
            ⊞ { } transaction_fee
              ■ receipt_id : "2257881548767648"
              ■ parent_payment : "PAY-5ND181847B062233PKYOT7EQ"
              ■ create_time : "2015-10-13T17:30:28Z"
              ■ update_time : "2015-10-13T17:30:31Z"
            ⊞ [ ] links
              ■ soft_descriptor : "PAYPAL *EMPRESATEST"
        ■ coupled_group : 0
    ■ create_time : "2015-10-13T17:30:31Z"
```

Handling errors on the execute payment API

For every error like bank decline, risk decline or malformed request, PayPal will return HTTP 400 and a JSON like the one below:

```
{
   "name":"TRANSACTION_REFUSED",
   "message":"The request was refused",
   "information_link":"https://developer.paypal.com/webapps/developer/docs/api/#TRANSA
CTION_REFUSED",
   "debug_id":"5c3b33b7d52d6"
}
```

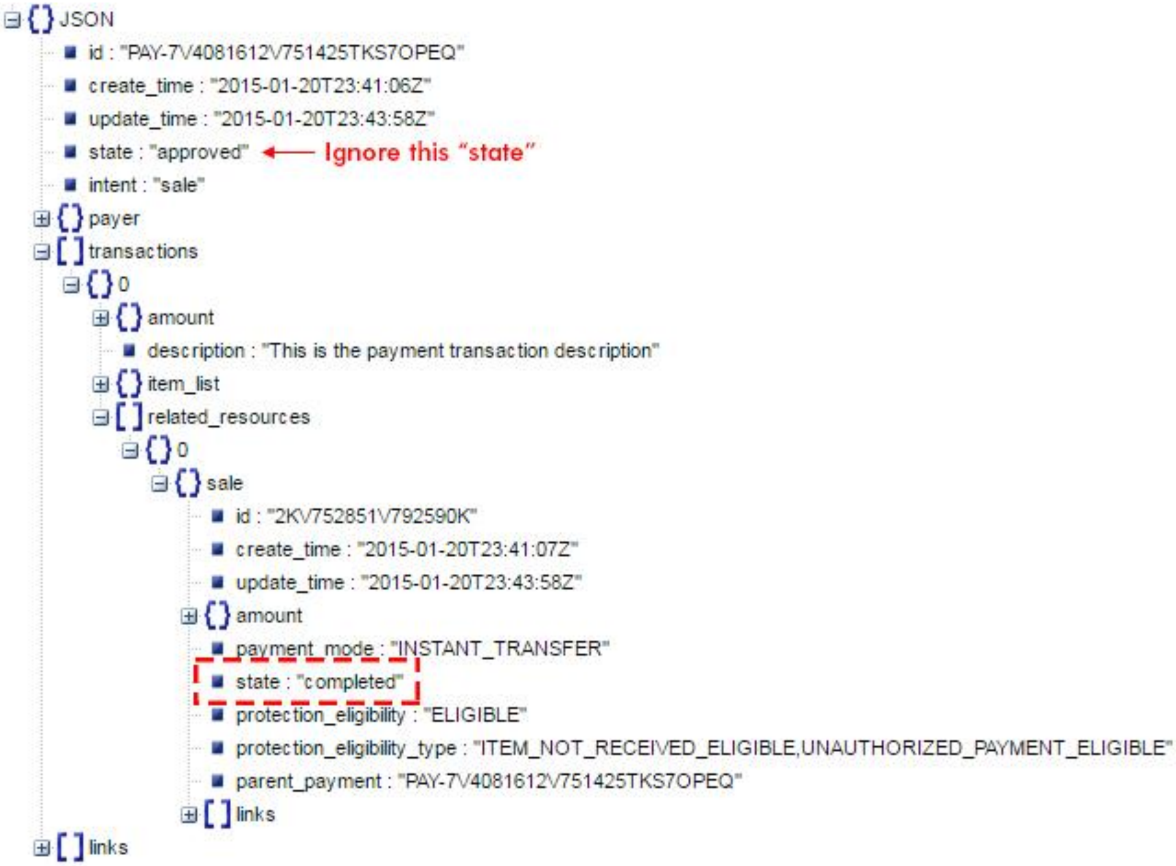
Use the “name” parameter to filter the real cause of the error and give the proper action, the most common errors are:

Name	**Action**
INTERNAL_SERVICE_ERROR	Call the API again with the same parameters and values.
INSTRUMENT_DECLINED	Bank decline, payment no approved.
CREDIT_CARD_REFUSED TRANSACTION_REFUSED_BY_PAYPAL_RISK PAYER_CANNOT_PAY PAYER_ACCOUNT_RESTRICTED PAYER_ACCOUNT_LOCKED_OR_CLOSED PAYEE_ACCOUNT_RESTRICTED TRANSACTION_REFUSED	PayPal risk decline, payment not approved.

Payment review

In some PayPal Plus integrations, payments can be reviewed by PayPal risk before fully approval. In this case PayPal will hold the payment for up to 24 hours before approving it or rejecting it. <u>Before continuing please check with your PayPal Account Manager if this functionality will be applied to your PayPal account.</u>

To identify if the payment will be reviewed by PayPal, check the Execute Payment API response and search for the "state" parameter inside the payload. If it's value is "completed" the payment is approved and is already revised by PayPal, no further action is required, but if it's "pending" this payment will be revised by PayPal, check the example below:



```
⊟ { } JSON
    ■ id : "PAY-7V4081612V751425TKS7OPEQ"
    ■ create_time : "2015-01-20T23:41:06Z"
    ■ update_time : "2015-01-20T23:43:58Z"
    ■ state : "approved"
    ■ intent : "sale"
  ⊞ { } payer
  ⊟ [ ] transactions
    ⊟ { } 0
      ⊞ { } amount
        ■ description : "This is the payment transaction description"
      ⊞ { } item_list
      ⊟ [ ] related_resources
        ⊟ { } 0
          ⊟ { } sale
              ■ id : "2KV752851V792590K"
              ■ create_time : "2015-01-20T23:41:07Z"
              ■ update_time : "2015-01-20T23:43:58Z"
            ⊞ { } amount
              ■ payment_mode : "INSTANT_TRANSFER"
              ■ state : "completed"
              ■ protection_eligibility : "ELIGIBLE"
              ■ protection_eligibility_type : "ITEM_NOT_RECEIVED_ELIGIBLE,UNAUTHORIZED_PAYMENT_ELIGIBLE"
              ■ parent_payment : "PAY-7V4081612V751425TKS7OPEQ"
            ⊞ [ ] links
  ⊞ [ ] links
```

Ignore this "state"

NOTE: There's another "state" parameter on the response, ignore it

Since PayPal is going to review the payment, you will need to integrate a service to retrieve the result once it's done, there're 2 methods, synchronous and asynchronous:

Service	**Description**	**For detailed instructions, see…**
Look Up API (Synchronous)	Look for a payment to check if it's already approved or rejected by PayPal, you will check the same "state" parameter on the API response. This method requires a routine to keep checking until the review is done.	Look for a Payment
IPN (Asynchronous)	Receive notifications from PayPal including payment reviews, you will need a callback service to receive POST's from PayPal with the payment status, this can also be used for refunds and disputes notifications.	IPN

Chapter 4. Useful REST API operations

Because PayPal Plus solution uses PayPal's RESTful APIs, you should also review and understand the following information before integrating:

How to…	**As described in PayPal REST API documentation…**
Use the sandbox environment for testing	API Operations
Issue requests to the sandbox and live environments	API Operations
Send request headers	Authentication & Headers
Refund a payment	Refund payments
Look up a refund	Look up a refund