

HPP Quick Start Guide

- [Introduction](#)
 - [Overview](#)
 - [Step 1: Get a list the HPP payment forms available in your organization](#)
 - [Headers:](#)
 - [Response Body:](#)
 - [Step 2: Generate the HPP payment form URL](#)
 - [Recommended Parameters:](#)
 - [Headers:](#)
 - [Request Body:](#)
 - [Response Body:](#)
 - [Step 3: Embed the generated URL into an iframe within your software application:](#)
 - [Step 4: Results and Reconciliation:](#)
 - [Results \(receipt\)](#)
 - [Reconciliation:](#)
 - [Callback_urls](#)
 - [How are callbacks used?](#)
 - [Redirects](#)
 - [Status Results](#)
 - [Payment Template - Zip Code](#)
 - [Hosted Payment Page Naming](#)
 - [Hosted Payment Page Template](#)

Introduction

The Hosted Payment Page can be added to your Bank of America Merchant Services Partner Account upon request. Once configured on your account by Bank of America Merchant Services, your development team can begin integration of the HPP payment form into your software solution. At its very minimum and basic configuration, the HPP payment form can often be implemented in just a few easy steps:

Overview

1. Make API GET call to list the HPP payment forms available in your organization
2. Make API POST call to generate the HPP payment form URL
3. Embed the payment URL into an iframe within your software solution to display the payment form and collect payment.
4. Reconciliation and results

Step 1: Get a list the HPP payment forms available in your organization

Request details:

URL: <https://settings-sandbox.paymentfusion.com/hpps>

Verb: GET

Headers:

- **Authorization:** Needs to include `Basic` word before Base64 encoded `api_id:api_token`. The `api_id` is the identifier of the organization created on TAP; size range - 20. The `api_token` is the unique token previously generated by TAP and provided securely to the organization; size range - 3
Example: `Basic ODA5BVgrNdZkZmbiFlJlTThkYTA2eG00MzA5dHh1bWZnZgfdZDE=`
- **Accept:** Specifies the requested API version. If this field is not specified or has an invalid format, the call will be routed to a previous version. If the previous version does not exist, a 404 page will be displayed.
Example: `application/vnd.paymentfusion.v1.0.0+json`
- **Content-Type:** `application/x-www-form-urlencoded`
- **Cache-Control:** `no-cache` Host: `settings-sandbox.paymentfusion.com`
- **Accept-Encoding:** `gzip, deflate`
- **Connection:** `keep-alive`

This will return a list of Hosted Payment Pages available to the organization, including the name of the payment form, as well as an identifier (corresponding to the HPP Source ID on our server). (Most organizations will return only one HPP source). There is no request body specified on this call. Based on the **Authorization** parameter (**API ID:API Token** – base64 encoded), the API call will know which HPP payment forms are associated with your organization.

Response Body:

```

{
  "hpps" : [
    {
      "name": "hpp_paymentform",
      "identifier": "6935274"
    }
  ],
  "v": "1.0.0"
}

```

The **identifier** is the field value that you will pass in to the **id** parameter on the next API call to generate the Hosted Payment Page payment URL link.

Step 2: Generate the HPP payment form URL

Request details:

URL: <https://tb-sandbox.paymentfusion.com/hpps>

Verb: POST

Required parameters:

The **transaction_type**, **amount**, and **id** parameters listed below are required; however, it is highly recommended to include the **approved_redirect** and the **declined_redirect** parameters because if they are not specified, then a default response template with limited formatting will serve as a "catch-all" webpage and may not be informative enough for your end-users

id= The HPP identifier for the transaction. This is the identifier field value returned in Step #1 and is unique to your organization Example: 6935274

transaction_type= The type of transaction that will be performed. Enum Auth, Sale, SaveOnly

amount= The amount of the transaction that will be performed. Example: 15.33

Note: If using the SaveOnly feature the amount has to be \$0.00.

Other input parameters exist here: <https://axiamed.atlassian.net/wiki/spaces/INP/pages/284624432/Hosted+Payment+Page> the partner portal also contains a link to this document.

The response from this call will return a dynamically generated, single use payment URL (see below).

Recommended Parameters:

- **callback_url_[1|2|3]**=One of the three URLs where Transaction Broker will POST to when a HPP request is completed. This eliminates the need for polling for the request's status. Only HTTPS URLs are allowed. The callback will retry if the host is unresponsive. The response received from the callback url must have the response code 2xx. Basic auth credentials may be added in to url ie. 'https://username:password@address.com' or as an url param '<https://address.com?token=secret>'. *Please note the use of approved or declined redirect parameters is the alternative to using the callback_url parameters.*
- **remote_id**= Used for keeping track of a 3rd party identifier alongside the transaction object. This is typically used by the ISV to assign its own identifier to the transaction for easy matching in case any network issues appear. This is not validated for uniqueness.

Headers:

- **Authorization:** Needs to include Basic word before Base64 encoded `api_id:api_token`. The `api_id` is the identifier of the organization created on TAP; size range - 20. The `api_token` is the unique token previously generated by TAP and provided securely to the organization; size range - 3
Example: `Basic ODA5BVgrNdZkZmbiFlJlTThkYTA2eG00MzA5dHh1bWZnZgfdZDE=`
- **Accept:** Specifies the requested API version. If this field is not specified or has an invalid format, the call will be routed to a previous version. If the previous version does not exist, a 404 page will be displayed.
Example: `application/vnd.paymentfusion.v3.0.0+json`
- **Content-Type:** `application/x-www-form-urlencoded`
- **Cache-Control:** `no-cache` Host: `tb-sandbox.paymentfusion.com`
- **Accept-Encoding:** `gzip, deflate` Content-Length: 47

- **Connection:** keep-alive

Request Body:

```
{
  "id": "",
  "transaction_type": "Sale"
  "amount": "Sale"
  ...
  "callback_url_1": ""
  "callback_url_2": ""
  "callback_url_3": ""
}
```

Response Body:

```
{
  "hpp": "https://gateway-sandbox.paymentfusion.com/hpp/0efc926ab2
/payment",
  "v": "3.0.0"
}
```

This URL can be passed in as the src parameter in the <iframe> within your payment page as detailed in step 3.

Step 3: Embed the generated URL into an iframe within your software application:

```
<iframe
src=https://gateway-sandbox.paymentfusion.com/hpp/0efc926ab2/payment
height="700" width="420" frameborder="0">
</iframe>
```

Your solution's end user is expected to fill out the form and submit it.

Step 4: Results and Reconciliation:

Results (receipt)

There are certain fields that should be displayed to the customer on a confirmation page after a payment has been processed.

1. Merchant Name
2. Customer service phone number
3. Transaction Amount.
4. Web site address

Reconciliation:

There are two methods that can be used to reconcile a hosted payment page transaction and they depend on how you requested the HPP URL in step 2.

Callback_urls

If you provided one of the callback_url(s) in your request for a HPP URL you will receive the following notification that your end user has submitted a payment form:

Response details:

URL: callback_url[1|2|3] supplied in step 2.

Verb: POST

Message body:

```
{
  "id": "12312",
  "type": "hpp",
  "version": "3.0.0",
  "data": {
    "id": 12312,
    "amount": "30.00",
    "invoice_number": null,
    "ip_address": null,
    "user_name": null,
    "transaction_status": "approved",
    "token": "12072498e7849bbb5d4d68a24bfed0eb39813b",
    "card_type": "Visa",
    "card_expiration_date": "0921",
    "masked_card_number": "XXXXXXXXXXXX1111",
    "check_account_type": null,
    "check_account_number": null,
    "transaction_type": "Sale",
    "created_at": "2020-08-05 20:40:35 +0000",
    "gateway_identifier": "payment_fusion",
    "gateway_reference_number": "123123",
    "gateway_auth_code": "DEMO123",
    "merchant_id": "123412341234",
    "billing_address_1": null,
    "billing_address_2": null,
    "billing_city": null,
    "billing_zip": null,
    "billing_state": null,
    "comment": null,
    "description": null,
    "card_holder": "CardHolder Name",
    "auth_amount": "30.0",
    "remote_id": null,
    "input_method": null,
    "bin_data": {
      "bin": 411111,
      "card_brand": "VISA",
      "issuing_org": "JPMORGAN CHASE BANK, N.A.",
      "card_type": "CREDIT",
      "card_category": "",
      "issuing_country": "UNITED STATES",
```

```

        "issuing_country_code_a2": "US",
        "issuing_country_code_a3": "USA",
        "issuing_country_number": 840,
        "issuing_phone": "1-212-270-6000",
        "issuing_website": "HTTP://WWW.JPMORGANCHASE.COM",
        "pan_length": "16",
        "issued_entity": "",
        "is_regulated": "Y",
        "is_commercial": false,
        "is_healthcare": false
    },
    "custom_fields": {}
},
"hash":
"b022d798c34f70f2fdf8210ec70a6203707ae12cf3766113051814b35d66fd59"
}

```

If possible it is suggested to save the entire response but we realize that may not be feasible, in these instance the following parameters must be saved:

- id
- amount
- auth_amount
- transaction_status
- token
- card_type
- masked_card_number
- card_expiration_date
- remote_id

If your integration employs the use of custom fields, those should be saved too.

How are callbacks used?

- Retry policy
 - Callbacks will be retried up-to 3 times with fixed back off of 1min.
- Multiple webhook callbacks are expected as transaction would get updated
 - For Card transaction this can be due to data getting updated from different sources:
 - Terminal
 - Tateway
 - There are 3 bits of data that can come late:
 - signature
 - bin_data
 - emv_data
- For ACH transaction this can be due to transaction status update which can be up-to 60 days after the transaction date.
- **Recommendation:** Callbacks can be used by partners for variety of reasons so it is hard to recommend one way or the other without understanding all the use cases. Some partners use it to block user flow for receipt generation until callbacks are complete others just use it to update their system for transaction status in background. Below is recommendation based on those 2 different category of changes:
- Background Transaction Updates
 - If there are not timelines to the background updates, meaning there is no use case where partner would be sending email receipts, etc in background. It is recommended that transaction data is stored in some storage, every webhook response should result in updating the transaction in the storage.
 - Keeping corner cases and retry policy in mind, expect transaction updates for up-to 5 mins (default transaction timeout or whatever the custom configuration would be) after the transaction was initiated
- In-flow Transaction Updates
 - If user interaction is blocked (for receipt printing) then recommendation is to wait for adequate data needed for the use cases or timeout at ~5 mins.
 - For example: If user flow is blocked for receipt printing then depending on input method type different data might be needed.
 - If input method is swipe or manual input then the first webhook should give all the data needed to print the receipt.
 - If input method is dip or tap then wait for webhook until you get EMV data for printing the receipt.

- Timeout ~5mins (or configured transaction timeout value) regardless of the input method

Redirects

If you provided redirect URL in your request for a HPP URL, approved and declined, you will receive the following notification that your end user has submitted a payment form, the message format will be the same for both approved and decline workflows:

Response details:

URL: approved or declined redirect value(s) supplied in step 2.

Verb: GET

Message payload:

```
/?
EchoFields=True&InvoiceNumber=&TaxAmount=&TransactionType=Sale&MerchantT
ransId=&IIASVerified=&HealthcareAmount=&CumulativeAmount=&ClinicalAmount
=&CopayAmount=&DentalAmount=&PrescriptionAmount=&TransitAmount=&VisionAm
ount=&Description=&Hash=8dba94660abfb1846eaec26f23be9785aa4d542af5be2d33
67f3f8bd0cef06da&ReceiptTemplate=&SecCode=WEB&CardBrand=VISA&CardNumberM
asked=XXXX-XXXX-XXXX-1111&Amount=30.
04&PaymentType=Credit&tb_id=10175&tb_transaction_type=hpp&CardExpiration
=0921&Batch=&BatchRefNumber=0&BatchStatus=Open&IsDuplicate=False&PNRef=2
52867&ProcRefNum=252867&ResultMessage=Approved+-
+DEMO&ResultCode=0&Token=1213845650e9c1aa6b419f81cbd4f85cd5cd6c&AuthCode
=DEMO780&CardAvsResult=Issuer+did+not+perform+AVS&CardAvsResultCode=0&Ca
rdCvvResult=Service+Not+Available&CardCvvResultCode=U&ResponseHash=069ce
799c286bb1153c3ad1a06ffae15b5f8e7adf8a85d6b2729f5001170690f&Result=Appro
ved&CardAuthAmount=30.04
```

Name	Type	Sample Value	Only present if EchoFields=True
EchoFields	boolean	True	
InvoiceNumber	string	2349812321	
TaxAmount	decimal	3.09	X
TransactionType	string	Sale	
MerchantTransId	string	98134929872342	
IIASVerified	boolean	False	
HealthcareAmount	decimal	10.00	
CumulativeAmount	decimal	10.00	
ClinicalAmount	decimal	10.00	
CopayAmount	decimal	10.00	
DentalAmount	decimal	10.00	
PrescriptionAmount	decimal	10.00	
TransitAmount	decimal	10.00	
VisionAmount	decimal	10.00	
Description	string	Custom description provided.	X
Hash	string	8dba94660abfb1846eaec26f23be9785aa4d542af5be2d3367f3f8bd0cef06d	X
ReceiptTemplate	string		X
SecCode	string	WEB	

CardBrand	string	VISA	
CardNumberMasked	string	XXXX-XXXX-XXXX-1111	
Amount	decimal	30.04	X
PaymentType	string	Credit	X
remote_id	string	a98b98c273f412312	X
tb_id	string	10175	X
tb_transaction_type	string	hpp	X
CardExpiration	string	0921	X
Batch	string		
BatchRefNumber	string	0	
BatchStatus	string	Open	
IsDuplicate	boolean	False	
PNRef	string	252867	
ProcRefNum	string	252867	
ResultMessage	string	Approved+-+DEMO	
ResultCode	string	0	
Token	string	1213845650e9c1aa6b419f81cbd4f85cd5cd6c	
AuthCode	string	DEMO780	
CardAvsResult	string	Issuer+did+not+perform+AVS	
CardAvsResultCode	string	0	
CardCvvResult	string	Service+Not+Available	
CardCvvResultCode	string	U	
ResponseHash	string	069ce799c286bb1153c3ad1a06ffae15b5f8e7adf8a85d6b2729f5001170690f	
Result	string	Approved	
CardAuthAmount	decimal	30.04	

Which fields should be save if the whole message can not be persisted?

- Amount
- Result
- PNRef
- AuthCode
- Token
- CardBrand
- CardNumberMasked
- CardExpiration

Status Results

How do I determine if a payment was successful? When determining if a payment was successful you will want to look at the **transaction_status** message that is available in the response payload or the callback url payload. This status will tell you if the transaction has been approved, declined or if an error has occurred.

Payment Template - Zip Code

Zip Code is now a mandatory field when setting up the hosted payment page. This is due to Address Verification Service and helps lower your processing interchange rates.

Hosted Payment Page Naming

We highly recommend when implementing the hosted payment page that you create a naming convention in a way that you can target a specific hosted payment page if you have multiple merchants under your organization. The default naming convention will be set up for you as HPP_merchant_id under the name field. Once ready to move to production our implementation team will follow said naming convention when onboarding new Merchants.

Hosted Payment Page Template

We do offer a basic template for you to use when integrating with our Hosted Payment Page solution. If that does not work for your integration we also can host a page that is developed off of our template by you. There are some restrictions in which we do not allow for outside css and javascript. We request that all changes be inline to the html template.

That's it! You should now have a fully functional payment form within your software solution. The Bank of America Merchant Services HPP is a powerful and versatile payment processing product that provides compliance, speed, and flexibility with minimal effort.

Card Not Present Address Verification Services(AVS)

MC charges a penny, but that is much better than the alternative of a card not present transaction run without AVS, which often leads to a downgrade (higher rate on the transaction)

Please consult the [Hosted Payment Page Template](#) for a more comprehensive description of its features as well as detailed information on how to control the look and behavior. For more information or support on the Hosted Payment Page, please contact us at ConnectPlatformManagement@bofa.com.