API - PayPage

Public - Security Level 0

March 2022

REVISION HISTORY

Date	Version	Author(s)	Comments
5/20/16	2.0	C. Meaney	Reformatting
5/4/18	2.1	M. Billips	Added support of BRIC-based transactions
5/11/18	2.2	M. Billips	Added support of Visa Debt Repayment chapter.
3/1/22	2.3	M. Billips	Add INDUSTRY_TYPE tag

CONFIDENTIALITY STATEMENT

Contents

EPX API – PayPage Introduction

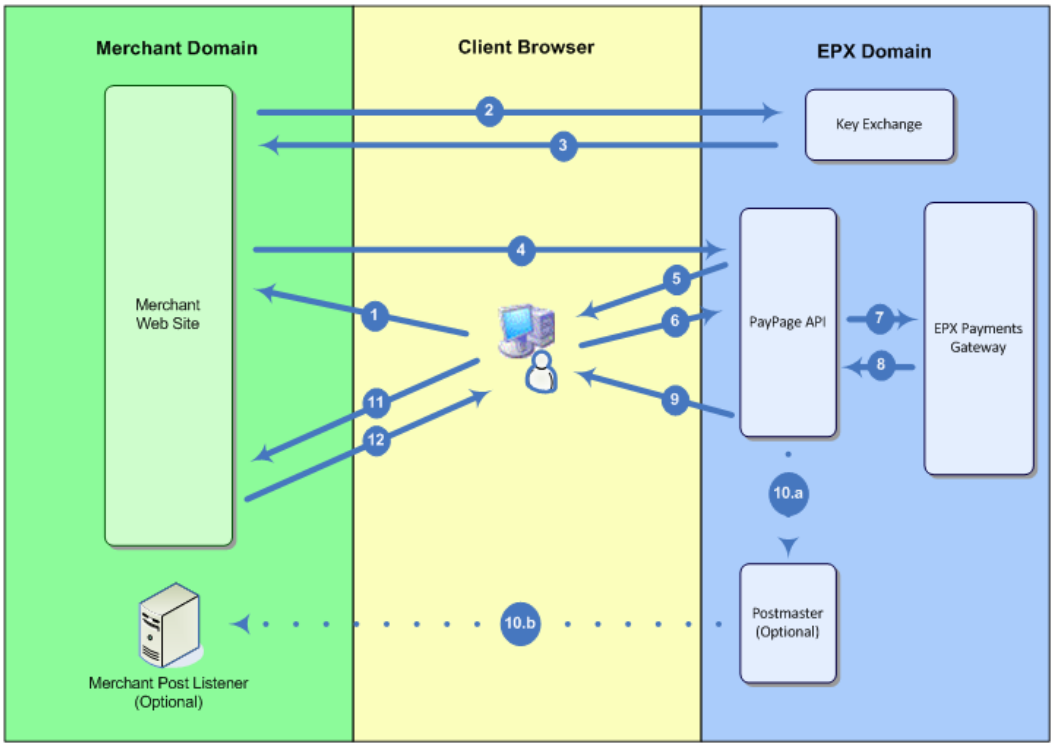
Overview

The EPX PayPage API is an http platform designed to allow the secure integration of an EPX-hosted payment page into a merchant-hosted web site. When implemented correctly, a merchant can call the fully customized PayPage API into a browser session which will then submit the transaction request to EPX for financial processing, with no critical card information ever existing on, or available to, the merchant site.

This is accomplished by configuring the checkout procedure to call the PayPage API when it is time to make payment, which will gather and post the customer and transaction information directly to EPX. After completion, the browser is redirected to the merchant-specified redirect URL on the merchant's site, and the results of the financial operation are also posted to the URL. Optionally, the API can also send the response to the EPX Postmaster which will post the response to an alternate URL where a merchant's listening service is available.

Process Flow

The figure that follows shows the processing flow for the EPX PayPage.



1. The customer signals the merchant site that he/she is ready to "check out" and make a financial payment. The final transaction amount should be available to the merchant at this time.

2. The merchant makes a TAC request to the KeyExchange service. This is an HTTPS request for a TAC (Terminal Authorization Code) token, to perform a transaction. The TAC is an encrypted value which contains merchant supplied information vital for transaction integrity and successful processing. This communication occurs directly between the merchant site and the KeyExchange service, so the client browser is never aware, and not privy, to this interaction.
3. The TAC is returned to the merchant site.
4. The merchant makes a PayPage request to the PayPage API. This is an HTTPS request that contains the TAC token, 4-part processing key, and can also include any merchant-provided information for the transaction, such as Invoice Number, User Data fields, and so on.
5. The PayPage is served from EPX to the client browser.
6. After entering their financial information, the user submits the PayPage from which will post the information back to the PayPage API.
7. The transaction is verified using a two-phase validation process. First, the TAC token is decrypted and all critical transaction information verified against the values posted to the PayPage API. Second, all posted fields are checked for potentially harmful content: cross-site scripting, SQL injection, and so on. If all validations are successful, a financial transaction is formatted and submitted to the EPX Payment Gateway for processing.
8. Financial transaction results are returned to the PayPage API.
9. The PayPage API redirects the client browser back to the merchant site, including the transaction response values.
10. (Optional) The response can also be sent to the EPX Postmaster for delivery to an alternate listening URL.
    a. PayPage API submits the transaction results to the EPX Postmaster application. In addition to the transactions results' availability in the browser redirection to the merchant site, a merchant may opt to receive the transaction response as an HTTP post to a central listening process.
    b. EPX Postmaster posts the transaction results to a central HTTP server in the merchant domain, not necessarily connected to the merchant web application site.
11. Transaction results are posted back to the merchant site.
12. Merchant returns a receipt page to the client browser.

KeyExchange Request

The KeyExchange process ensures transaction integrity and security by encrypting merchant-provided values into a token that is included in the subsequent submission to the PayPage API. This KeyExchange occurs between the merchant site and EPX, and cannot be accessed by the client browser in any way. Each and every post to the PayPage API must contain a unique TAC token.

There are several required fields that a merchant must provide in each TAC request to guarantee transaction integrity. The merchant can optionally provide the value for any valid field from the subset outlined in the *Ecommerce Regular Expression Reference* guide for inclusion in the TAC token. If that same field is provided in the POST that is later submitted to the PayPage API, it must match the TAC value during the PayPage API validation process. This is a safeguard against any man-in-the-middle tampering that could otherwise occur. Finally, there is a time component included in the TAC value which invalidates the token after a four-hour period.

Request Fields

Please refer to the *Data Dictionary* for more information on individual fields.

Required Fields

- MAC (Merchant Authentication Code)
- AMOUNT
- TRAN_NBR
- TRAN_GROUP

Example of KeyExchange Request

```
https://keytest.test.com/?tran_
nbr=12345678&amount=10.00&mac=pZCt1BnhTZSiXtLtX9t1II51ZP35sfus&tran_
group=SALE”
```

Example of KeyExchange Response

```
<RESPONSE>
<FIELDS>
<FIELD KEY="TAC">FEU86Eo5/S1CZwZemrks4P7w1IpJSFIi7qTQ+Sn
ygqi0jAAyC+7TYZq5wojO94pNE7gQDCjC1XMgo
</FIELDS>
</RESPONSE>
```

All requested fields are encrypted into the TAC token and returned to the merchant site, so that the token can then be included in the post by the web payment form to the PayPage API.

PayPage Request

The PayPage request should contain the required fields listed below. There are also a number of optional fields that can be used to configure various PayPage features, including those that configure the use of the Postmaster system.

Request Fields

Please refer to the *Data Dictionary* for more information on individual fields.

Field	Required	Optional
AMOUNT	X	
BATCH_ID	X	
CANCEL_URL		X
CUST_NBR	X	
CUSTOM_CSS_URL *Used during certification		X
CUSTOM_JAVASCRIPT_URL *Used during certification		X
CUSTOM_PRINT_CSS_URL *Used during certification		X
DBA_NBR	X	
FORCE_CREDIT		X
INDUSTRY_TYPE	X	
INVALID_REDIRECT_URL		X
MERCH_NBR	X	
RECEIPT		X
REDIRECT_ECHO		X
REDIRECT_NON_APPROVALS		X
REDIRECT_URL	X	
RESPONSE_ECHO		X

Field	Required	Optional
RESPONSE_FORMAT		X
RESPONSE_URL		X
TAC	X	
TERMINAL_NBR	X	
TRAN_NBR	X	

The merchant can optionally populate any field from the subset outlined in the *Ecommerce Regular Expression Reference* guide during their post in order to pre-populate data in the page, or include additional information in the background to be part of the transaction.

For an example of a PayPage request implementation, please see the provided PayPage Demo code.

PayPage API

Upon receipt of a POST, the PayPage API goes through a series of three processes: Validation, Payment Processing, and Response Reporting.

Validation

The validation process begins with decrypting the TAC token received in the PayPage Request. First, the process checks for all required fields in the TAC. The process then compares the TRAN_CODE field received in the POST to the TRAN_GROUP for compatibility. Next, to ensure that no tampering occurs with the POSTed form fields, the process compares any fields supplied in the TAC to the POSTed values for equality.

IMPORTANT

All EPX API fields that are submitted to the EPX Payment Gateway are taken from the POSTed form fields and not the TAC token. For example, if a merchant supplies the INVOICE_NBR in the TAC token, but not in the web form, no value for the INVOICE_NBR field is submitted to the EPX Payment Gateway for the transaction. If the INVOICE_ NBR field is contained in the PayPage Request, but not the TAC, it is submitted, as is, to the EPX Payment Gateway. If the INVOICE_NBR is contained in both the TAC and the POST, the values must match or the transaction will fail PayPage API validation. Finally, each submitted EPX API field that is accepted by the PayPage API must pass a validation test, or else the entire transaction will fail the PayPage API validation. The field validations are controlled by a Regular Expression test that is outlined in the *Ecommerce Regular Expression* reference guide.

If a transaction fails any validation, the PayPage API immediately redirects the browser to back to the PayPage without submitting the transaction to the EPX Payment Gateway for financial processing. If this occurs, a general error message is displayed indicating why the redirect occurred. In addition, the PayPage contains all POSTed values to the PayPage API, so that the consumer can correct any visible fields that have an issue and then have the opportunity to resubmit the transaction.

Payment Processing

If a transaction passes all PayPage API validations, an EPX Payment Gateway transaction is formatted and submitted for financial payment processing. All accepted fields are extracted from the PayPage API POST and included in the transaction request. The PayPage API passes the financial transaction response to the Receipt page. Once the consumer clicks the continue button on the Receipt page, the PayPage API passes the financial transaction response as a POST to the specified REDIRECT_URL.

At a minimum, the AUTH_RESP should be checked for the success or failure of the transaction. If preferred, the RECEIPT field can be sent during the transaction to disable the default Receipt page and allow the consumer to be immediately redirected to the REDIRECT_URL where a merchant-designed receipt page is available to receive the response. For complete definitions for all fields mentioned please refer to the *EPX Data Dictionary*.

Response Reporting

After a transaction is processed by the PayPage API, the response is formatted and returned to the client browser as a self-posting form to the URL specified in the REDIRECT_URL. This ensures that the transaction response is POSTed back to the merchant site and available for merchant data processing before any results are made visible to the customer in the client browser. Typically, the merchant takes this opportunity to insert the results to a database, and formats and returns a receipt page with any fields from the transaction response, if approved. If declined, the merchant must take the appropriate action.

Finally, a merchant can optionally request that all financial transaction responses (EPX Payment Gateway) are POSTed to a merchant listener for the response POST. This is done by including values for the RESPONSE_ URL, and optionally the RESPONSE_FORMAT and RESPONSE_ECHO fields in the PayPage Request.

Please refer to the *EPX Reference - PostMaster Response* for more information.

PayPage Customization/Branding

You can customize the PayPage look and feel to provide a seamless customer experience between the merchant site and the hosted payment page. JavaScript and CSS files control the appearance and the behavior of the PayPage. Contact an EPX Integration Specialist for sample files.

WARNING!

Iframe Implementation - Because the PayPage is hosted at an EPX URL, requesting the page to display in an iframe window causes the PayPage to be subjected to the 3rd Party security settings of the client browser. This is due to the iframe PayPage URL, epx.com, not matching the URL of the parent page where it is hosted, merchantsite.com. The results can range from the page not being displayed correctly, to the page appearing normal, but not processing the transaction correctly due to session variables not being set as needed. Because it is difficult or sometimes impossible for the merchant or EPX to manage the end users browser settings, we insist that the best practice is to allow the PayPage to be a full page in the checkout procedure. As described below, the PayPage is fully customizable allowing seamless transition through the checkout for the customer. Implementing the PayPage in this manner ensures that it will consistently work correctly for the end user once deployed in production.

EPX PayPage uses JavaScript and CSS to allow customization and manipulation of the hosted PayPage code, enabling the merchant to achieve specific visual and business requirements. The scripts enable you to show and hide fields, adjust table sizes, and add html or images to fully customize the page and replicate the look and feel of the merchant site. The JavaScript can also be used to trigger functionality or customization dynamically using conditional statements that review data received during the PayPage Request.

By default, the PayPage displays radio buttons for each payment type allowed on the PayPage terminal. You can override these buttons if a specific payment type is desired for use on the page. To set a specific payment type on the page the TRAN_CODE field is used. The integration package contains sample scripts for the various transaction types that can be forced through the PayPage in this manner. The flexibility in the JavaScript can also allow the use of more than one TRAN_CODE using conditional statements that set the value, allowing the transaction type to be set dynamically when the page is called. It is important that the TRAN_GROUP in the KeyExchange request still corresponds with the TRAN_CODE used. Please refer to the *EPX Data Dictionary* for more information on the TRAN_CODE and TRAN_GROUP fields.

During integration, the CUSTOM_CSS_URL and CUSTOM_JAVASCRIPT_URL can be used to identify a locally hosted version of the scripts allowing for easier development of the customization.

Once you are satisfied with the changes, submit the files to an EPX Integration Specialist. Internally, they are audited according to established PCI and debit bill payment guidelines to

verify that no insecurities or account holder data interactions have been introduced. For more information relating to PCI requirements, please refer to http://www.PCISecurityStandards.org

If debit bill payment is used, the guidelines include the following requirements:

1. User Authentication—This is outside of the scope of the PayPage, but the consumer should be authenticated.
2. Bill Payment Selections—The consumer must be offered a choice as to the type of payment and/or payment instrument to be made (ACH/Debit Card/Credit Card), to the extent that the merchant offers those payment types.
    a. Merchants may not convert a Visa or MasterCard payment to a debit payment without consumer consent.
    b. Merchants may prompt or encourage consumers to use one form of payment over another, but must give the consumer an option to choose his or her preferred payment.
3. Logos—Debit network and credit network logos must be present, to the extent that the merchant offers those payment types.
4. Debit Bill Payment Disclaimer—If debit is offered, language must be present to the following effect: "Note: If you make payment with a debit card, the Payment amount will be immediately deducted from your checking account. This deduction amount cannot be reversed or voided upon deduction from your account."
5. Decline Message—If a debit transaction is attempted and the card is not eligible for debit bill payment, a message should be displayed stating such. For example: Card declined. No card on record.

Once they pass this review, the script and images files are hosted in the EPX production environment at the DBA level in the merchant's hierarchy. The CUSTOM tags are ignored in the production environment. It is a requirement that at minimum the default customization scripts are edited for a proper title and provided to integration for implementation in production.

Support of BRIC-Based Transactions

PayPage supports the use of BRIC-based transactions. This is accomplished by sending the ORIG_AUTH_GUID tag with the desired EPX BRIC in the PayPage POST.

IMPORTANT

Both the ACCOUNT_NBR and EXP_DATE tags must be omitted for the transaction to process with the BRIC.

Following are the supported TRAN_CODE(s) for BRIC-based transactions:

- AUTH
- AVS
- SALE
- STORAGE

Support of Visa Debt Repayment

PayPage supports Visa Debt Repayment (Sale) transactions. This is accomplished by sending TRAN_CODE of "BILLPAY" in the PayPage POST. The PayPage API will format the transaction request appropriately for a successful transmission. For detailed information on Visa Debt Repayment transactions, reference *EPX Transaction Specs - Visa Debt Repayment*, and for TRAN_CODE details, reference the *EPX DATA Dictionary*.

IMPORTANT

- The production terminal profile must be configured for Ecommerce. This transaction type is not supported for any other industry type.
- The client application is responsible for determining that the Card Type is Visa prior to the submission of the transaction, as this transaction type is specific to Visa Debit only.