API - Browser Post

Public - Security Level 0

March 2022

REVISION HISTORY

Date	Version	Author(s)	Comments
5/20/16	2.0	C. Meaney	Reformatting
3/1/22	2.1	M. Billips	Add INDUSTRY_TYPE tag

CONFIDENTIALITY STATEMENT

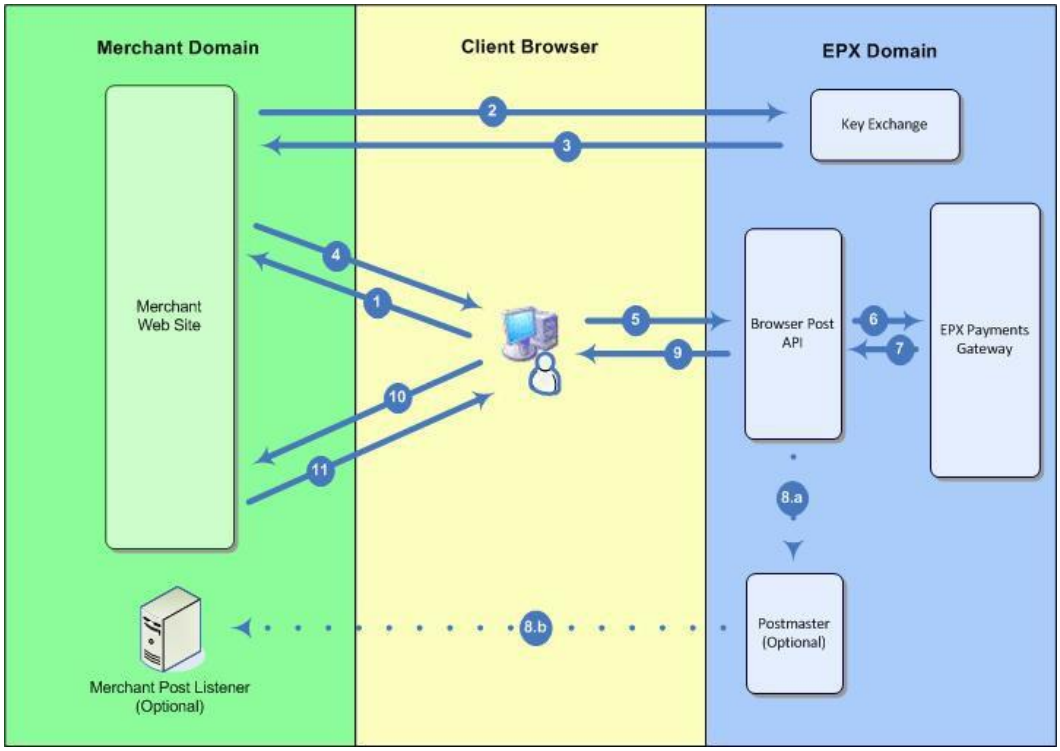
Contents

EPX API - Browser Post Overview

The EPX Browser Post API is an http platform designed to allow the secure integration of financial transactions from a merchant-hosted website. When implemented correctly, a merchant can serve a web payment form to a browser and 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 payment page to post the cardholder and transaction information directly to the Browser Post API for financial processing. After completion, the browser is redirected to the merchant site with the results of the financial operation.

Process Flow

The figure that follows shows the EPX Browser Post process flow.



1. The end user 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 EPX Key Exchange service. This is an HTTPS request for a TAC (Terminal Authorization Code) token, to perform a transaction. The TAC is an encrypted value that contains merchant supplied information vital for transaction integrity and successful processing. This communication occurs directly between the

merchant site and the EPX Key Exchange service, so the client browser is never aware of, and not privy to, this interaction.

3. The TAC is returned to the merchant site.
4. The merchant serves a payment page to the client browser. This form should POST directly to the Browser Post API. The form must contain the TAC token, and may also include any merchant provided information for the transaction, e.g. Invoice Number, User Data fields etc.
5. After entering their financial information, the user submits the web payment form directly to the Browser Post API.
6. 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 Browser Post API. Second, all posted fields are checked for potentially harmful content: cross-site scripting, SQL injection, etc. If all validations are successful, a financial transaction is formatted and submitted to the EPX Payments Gateway for processing.
7. Financial transaction results are returned to the Browser Post API.
8. (Optional) The response can be sent to the EPX Postmaster for delivery to an alternate listening URL.
    a. Browser Post 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.
9. The Browser Post API redirects the client browser back to the merchant site, including the transaction response values.
10. Transaction results are posted back to the merchant site.
11. Merchant returns a receipt page to the client browser.

Key Exchange Request

The key exchange processes ensures transaction integrity and security by encrypting merchant provided values into a TAC (Terminal Authentication Code) included in the form post for a subsequent submission to the Browser Post API. This key exchange occurs between the merchant site and EPX, and cannot be accessed by the client browser in any way. Each and every post to the Browser Post 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 a subset of the entire EPX Payment API for inclusion in the TAC token. If that same field is provided in the payment form that is later submitted to the Browser Post API, it is required to match the TAC value during the Browser Post 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 will invalidate the token after a four-hour period.

Request Fields

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

Field	Required	Optional
MAC (Merchant Authorization Code)	X	
REDIRECT_URL	X	
AMOUNT	X	
TRAN_NBR	X	
TRAN_GROUP	X	
REDIRECT_ECHO		X
RESPONSE_URL		X
RESPONSE_FORMAT		X
RESPONSE_ECHO		X

NOTE: *When used, optional TAC fields are read from the decrypted TAC and not from the web form post.*

Example of Key Exchange Request

```
https://testkey.test.com/?tran_
nbr=12345678&amount=10.00&mac=pZCt1KnhTZZiXtLtX9t1PP51AR35sfus&
tran_group=SALE&redirect_url=http://localhost/browserpost/
browserposttest.asp
```

Example of Key Exchange Response

```
<RESPONSE>
<FIELDS>
<FIELD KEY="TAC">FEU86Eo5/S1CZwZemrks4P7w1IpJSFIi7qTQ+sNYgqi0jAAyC+
7GDOq5wojO94pNE7gQDCjClXMgo+Gez9GBgSAQXaF/rX7J</FIELD>
</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 Browser Post API.

Web Payment Form

The web payment form is designed by the merchant, but there are some general requirements which must be implemented for successful transaction processing through the Browser Post API:

- The form must POST directly to the Browser Post API.
- The form must include a unique TAC value for each POST to the Browser Post API.
- The form must include the TRAN_CODE field, which corresponds to the EPX API TRAN_TYPE field. Please refer to the *Data Dictionary* for a full field definition.
- The form must include the INDUSTRY_TYPE field with a value of "E" to indicate an industry type of Ecommerce. Please refer to the *Data Dictionary* for a full field definition.
- The form must include the following EPX API fields:
  - CUST_NBR
  - MERCH_NBR
  - DBA_NBR
  - TERMINAL_NBR
  - AMOUNT

The merchant can optionally populate any field from a subset of the entire EPX Payment API.

For an example of a web payment form implementation, see the provided Browser Post Demo code.

Browser Post API

Upon receipt of a POST, the Browser Post 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 Browser Post API. First, the process checks for all required fields in the TAC. It is especially critical that a valid TAC token is received, and that the REDIRECT_URL is contained therein. Otherwise, the Browser Post API is unable to redirect the client browser back to the merchant site. The domain portion of the REDIRECT_URL and the INVALID_ REDIRECT_URL should be stored at EPX to assist with the validation process. 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.

NOTE: All fields that are submitted to the Browser Post API are taken from the POSTed form fields and not the TAC token, unless explicitly mentioned in the KeyExchange section. 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 Browser Post API for the transaction. If the INVOICE_NBR field is contained in the Browser Post API, but not the TAC, it is submitted, as is, to the Payment Gateway. If the INVOICE_NBR is contained in both the TAC and the POST, the values must match or the transaction will fail Browser Post API validation

Finally, each submitted EPX API field that is accepted by the Browser Post API must pass a validation test, or else the entire transaction will fail the Browser Post 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 Browser Post API immediately redirects the browser to REDIRECT_ URL without submitting the transaction to the EPX Payments Gateway for financial processing. If this occurs, the Browser Post API includes two response tags: BP_RESP_CODE and BP_RESP_TEXT. In addition, the response contains all POSTed values to the Browser Post API, so that the merchant site can re-present the web payment page to the user and prompt him/her for corrections, with the exception of the fields containing financial card holder data. The user will need to re-enter all card information. Finally, if the validation failure is the result of a field entry regex failure, the Browser Post API returns a BP_RESP_CODE value of BP_140, and includes a response tag named BP_FIELD_ERRORS. For complete definitions for all fields mentioned, refer to the *EPX Data Dictionary*.

Payment Processing

If all Browser Post API validations are passed, an EPX Payment Gateway transaction is formatted and submitted for financial payment processing. All accepted fields are extracted from the Browser Post API POST and included in the transaction request. The Browser Post API passes along 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 a transaction is submitted to the Payment Gateway, the BP_RESP_CODE and RTPB_RESP_TEXT fields will not exist in the transaction response. For complete definitions for all fields mentioned, refer to the *EPX Data Dictionary*.

Response Reporting

After the Browser Post API processes a transaction, the API formats the response and returns it 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 made available for merchant data processing before any results are displayed to the end user in the client browser. If the request fails Browser Post API validation, the BP_RESP_CODE and BP_RESP_TEXT fields are available in the request, along with all POSTed values, so that the merchant can re-populate and display the web payment page to the end user. Else, the financial transaction response is made available. Typically, the merchant will take this opportunity to insert the results to a database, and format and return a receipt page with any fields from the transaction response, if approved. If declined, the merchant must take the appropriate action, i.e. – prompt for a different method of payment, decline the transaction, etc.

NOTE: EPX cannot be responsible for browser behavior when the "BACK" or "REFRESH" buttons are clicked in the client browser after a response is received. For some browsers, this results in a repeat of the transaction request to EPX. To prevent multiple financial transactions processing in response to duplicate POSTs, the Browser Post API implements the PRG (POST-REDIRECT-GET) paradigm. In this model the transaction is POSTed to EPX for processing once. After the transaction has been processed the browser is redirected to a response page on the EPX Browser Post API site, where the browser is instructed to GET the transaction response. Any duplicate POSTS or re-requests for the GET (transaction response) operation result in the redirection of the browser to the Browser Post API response page, which, in turn, redirects the browser to the REDIRECT_URL on the merchant site with the transaction response. The merchant site, therefore, must be prepared to handle the situation where a response for the same transaction can be received multiple times. One possible method a merchant can use could involve a lookup on the BRIC/GUID response prior to insertion in the merchant database.

Finally, a merchant can optionally request that all financial transaction responses (EPX Payment Gateway) are POSTed to a merchant listener for the response POST. To do so, include values for the RESPONSE_ URL, and optionally the RESPONSE_FORMAT and RESPONSE_ECHO

fields in the Key Exchange request. Refer to the *EPX Reference - PostMaster Response* guide for more information.