BlueFire Donations

Sale API Interface Definition

The BlueFire Sale API interface allows you to capture payment information on your own website and pass it through your server to the BlueFire endpoint (server-to-server) and verify that the payment is complete.

Please note that because of the nature of E-Check transactions, an order may immediately return a success indication and later be rejected (e.g., due to non-sufficient funds, an incorrect account number, etc). If you are accepting E-Check payment for digital or physical goods we recommend waiting a week before shipping to verify that the payment was successful. Refer to the Reporting API for info on how to check the status of a transaction.

Endpoints

URI: https://api.bluefire-secure.com/sale

GET Parameters

Name	Type	Default	Description
key	String	None (field req.)	API access key, available upon request.
format	['JSON']	'JSON'	Response format (JSON only available option at this time).
test	['true','false']	'false'	Test transaction.

POST Parameters

Name	Type	Default	Description
donation	[‘true’,’false’]	‘false’	This transaction is a donation.
paymentType	[‘ach’,’credit’,’ swipe’,‘user’]	None (field req.)	The type of payment.
ecRout	String (9 digits)	None (req. for ACH)	The e-check routing number.
ecAcct	String	None (req. for ACH)	The e-check account number.
ecType	[‘C’,’S’]	None (req. for ACH)	Checking or savings account.
ccNum	String	None (req. for CC)	[2]The RSA-encrypted credit/ debit card number (ie, PAN).
ccExpM	String (2 digits)	None (req. for CC)	The credit/debit card expiration month (Between 01 and 12).
ccExpY	String (2 digits)	None (req. for CC)	The credit/debit card expiration year.
ccCVV2	String (3 or 4 digits)	None	The credit/debit card security code. Optional for donations.
track1	String	None (req. for swipe)	Raw track 1 data from credit card swipe.
track2	String	None (req. for swipe)	Raw track 2 data from credit card swipe.
userPaymentToken	String	None (req. for user)	The user’s payment method token (returned by User API as paymentKey).
userToken	String	None (req. for user paymentType)	The user’s oAuth access_token (see oAuth documentation for information on how to retrieve this). Indicates which user to associate this payment with.
title	String	None	Donor’s title. Required for accounts with this option enabled. Must match account’s predefined options.
firstName	String	None (field req.)	Billing contact’s first name.
lastName	String	None (field req.)	Billing contact’s last name.
employer	String	None	Billing contact’s employer. Not used for most accounts.

Name	Type	Default	Description
address1	String	None	Billing contact's street address.
address2	String	None	Line 2 of the billing contact's street address.
city	String	None	Billing contact's city or locality.
state	String	None	Billing contact's state or province.
zip	String	None (field req.)	Billing contact's zip or postal code.
country	['US', 'CA', 'MX', …]	'US'	Billing contact's country. Use ISO 3166-1 alpha-2 abbreviations.
phone	String (10 digits for US & CA)	None	Billing contact's phone number.
email	String	None (field req. for recurring trans.)	Billing contact's email address.
memberId	String	None	Member's ID, if applicable.
comments	String	None	Additional transaction comments. (Will appear on receipts. Use for line breaks. No other HTML is valid.)
invoiceNo	String	None	Invoice number, if applicable. Not available on donations.
fund	String	None (field req.)	[1]The fund or designation for the funds. (Will appear on receipts.)
fundNumber	String	None	[1]The associated fund numbers with the above funds. (For reporting, not included on receipts.)
fundRecurr	['One-time', 'Weekly', 'Bi-weekly', 'Semi-monthly', 'Monthly', 'Quarterly']	'One-time'	[1][3]The associated recurring period for the funds specified. When set do not use the 'recurring' field.

Name	Type	Default	Description
amount	Float	None (field req.)	[1]Amount of charge.
recurring	[‘true’, ’false’]	‘false’	[3]Set up a recurring transaction. Must be enabled for your account. When set do not use the ‘fundRecurr’ field.
startOn	String (mm/dd/yyyy)	None	Recurring transaction start date.
endDateSet	[‘true’, ‘false’]	‘false’	Recurring transaction perpetuity.
endOn	String (mm/dd/yyyy)	None	Recurring transaction end date (if applicable).
period	[‘Weekly’, ‘Bi-weekly’, ‘Semi-monthly’, ‘Monthly’, ‘Quarterly’, ‘Annually’]	None	Recurring interval.
receiptNote	String	None	A note to include on email receipts, at the bottom just before the organization’s address. (Use for line breaks. No other HTML is valid.)
emailReceipt	[‘true’, ‘false’]	‘true’	Email a receipt to the billing contact (if an email address is provided). Forced ‘true’ for recurring transactions.
remoteIp	String	None (field req.)	Remote client’s IP address.

[1]Multiple Funds/Line Items

Multiple funds (with optional fund numbers) and amounts can be specified by using an array format: Set each fund with the parameter “fund[]”, each corresponding fund number with the (optional) parameter “fundnumber[]” and a dollar amount with the parameter “amount[].” For example, the encoded POST data may contain a section that appears as follows:

```
fund[]=Wall%20Poster&amount[]=20.00&fund[]=Music%20CD&amount[]=25.00
```

[2]Card Account Encryption

Card account numbers must be encrypted client-side using the BlueFire JavaScript RSA encryption library before the data is submitted. This requirement reduces your Payment Card Industry Data Security Standards (PCI-DSS) scope, as payment information is fully encrypted while being transmitted by your server.

Include the following script in your client-facing web page during checkout. (Please contact BlueFire to ensure that you are using the latest version of this script as this document may be out of date.)

```
https://bluefire-secure.com/crypto/js/bf-encrypt-latest.min.js
```

Before submission, or as is appropriate in your application, encrypt the credit card information and mask the unencrypted data so it is not passed to your server. The following two functions are provided to perform these tasks:

```
// bf_encrypt() returns an encrypted version of the CC number
// This will be passed to BlueFire
function bf_encrypt(cardnumber);

// bf_mask() returns a masked CC number (maskchar = 'x' by default)
function bf_mask(cardnumber, maskchar);
```

[3]Recurring Options

There are two ways to set up recurring transactions. Only one method can be used per transaction.

The first way is to set up recurring on a fund-by-fund basis by passing the recurring period in the "fundRecurr" field. This field accepts one of the following values: "One-time", "Weekly", "Bi-weekly", "Monthly" and "Quarterly". This method of setting up recurring donations and payments useful when mixing recurring with non-recurring transactions into a single transaction. This method of setting up recurring transactions does not have more advanced options available such as future start date or end date. Note [1] does apply to this field and should have as many options as the "fund" field.

The second way of setting up a recurring transaction is to set the entire transaction (all funds, as a group) to recur. To do this set the "recurring" field to "true". Additional parameters are available, as detailed in the parameters table, including "startDate", "endDateSet", "endDate" and "period". Note that, in contrast to the method explained in the previous paragraph, these recurring settings apply to all funds that are a part of the transaction.

Response Data

Response data is encoded in the format specified. There are three parameters returned:

Field	Description
status	Either ‘1’ for success or ‘0’ for failure.
transactionId	The ID assigned to this transaction, if successful. Will be 0 if failed or if it is a future recurring transaction.
error	The textual error message, if applicable.
errorNo	The corresponding error number. 0 = No error 1 = Access error 10 = Generic field error (eg, invalid first name) 20 = Payment configuration error 30 = Payment configuration error 100 = Payment declined 101 = Payment error 102 = Payment error (no response received)