

August 16, 2019



# Integration Guide

---

**v1.9**

---

## Contents

Contents.....	2
Change Log.....	3
Integration Files .....	5
Calling the CashEnvoy Payment Interface .....	7
Retrieving Payment Response .....	9
Refund (Wallet Payments Only) .....	9
Broken Transactions .....	10
Recurrent Transactions – Taking Future Payments.....	11
Recurrent Transactions – Refund .....	12
Recurrent Transactions – Cancel .....	13
Additional Information .....	14
Additional Notes .....	15

## Change Log

16/08/2019 [v1.9]

- Added IPN support for non-recurrent transactions. IPN will be sent if variable ce\_ipnurl is sent when calling the payment interface.

15/01/2018 [v1.8]

- Updated test environment URLs.
- Added interface to refund of wallet transaction.
- Added test card details for local and international transactions to response code and test accounts document.

30/07/2017 [v1.7]

- Updated documentation to cover updates to recurrent transactions.
- Variable named ce\_recurrent can now be set to “none” when calling the payment interface for recurrent transactions. Transactions in payment cycles created this way do not need to have the same amount.
- Added interface to allow retrieval of card basic card data associated with a recurrent transaction.

01/05/2017 [v1.6]

- Updated documentation to cover updates to recurrent transactions. Sections updated are: [Integration Files](#), [Calling the CashEnvoy Payment Interface](#), [Additional Information](#) and [Recurrent Transactions](#).
- New variable ce\_cyclemgt to be sent when calling the payment interface for recurrent transactions.

02/02/2017 [v1.5]

- Updated documentation to cover recurrent transactions. Sections updated are: [Integration Files](#), [Calling the CashEnvoy Payment Interface](#), [Additional Notes](#) and [Recurrent Transactions - Explained](#).
- New variables ce\_recurrent and ce\_ipnurl to be sent when calling the payment interface for recurrent transactions. Transactions in payment cycles must have the same amount.

28/08/2014 [v1.4]

- Updated the “Retrieving payment response” with information on preventing man in the middle attacks.

14/08/2014 [v1.3]

- New variable ce\_signature to be sent when calling the payment interface.

09/04/2012 [v1.2]

- Transaction status response, obtained via requery, can now be returned in xml or json. Previously, only the string option was available. The example\_requery\_usage.php file has been modified to show how this can be done.

21/05/2011 [v1.1]

- C06 and C07 are no longer in use but merchants do not need to edit old codes. Nothing will be broken due to this change.
- The response codes and test accounts file has been updated in line with the above.
- Retrieving payment response via post has now been deprecated. However, those still using this method will have till the 1<sup>st</sup> of July 2011 to upgrade. The file example\_capture\_response is no longer a part of the integration pack.
- When calling the payment interface, the variable ce\_data is no longer necessary since all transaction responses will now obtained as shown in the file example\_requery\_usage. The sample code in this file or any other preferred methods should be used within the notification URL page to retrieve transaction status.
- A minor change has been made in the description of the parameters needed to call the payment interface within this manual.
- Minor changes have also been made to the example\_callcepayment\_interface file accordingly.
- An additional information section has been added to the end of the manual. It contains some extra information regarding going live.

## Integration Files

The integration package contains the following documents namely:

- 1.) example\_callcepayment\_interface.php - this file gives an example of how to call the payment interface.
- 2.) example\_callcepayment\_interface\_rc.php - this file gives an example of how to call the payment interface to setup a recurrent transaction.
- 3.) example\_callcepayment\_interface\_rc2.php - this file gives an example of how to call the payment interface to setup a recurrent transaction **with no defined frequency**.
- 4.) example\_requery\_usage.php - shows an example of how to get the status of a previous transaction via the API. .
- 5.) example\_ipn\_usage.php – shows an example of how to process transaction data posted to your specified IPN URL.
- 6.) example\_recurrenttx\_pay.php – shows an example of how to trigger a subsequent transaction in a payment cycle with defined frequency. **Note that in this case, all transactions in a payment cycle will have the same amount.**
- 7.) example\_recurrenttx\_pay2.php – shows an example of how to trigger a subsequent transaction in a payment cycle **with no defined frequency. Note that in this case, each transaction in a payment cycle can have a different amount.**
- 8.) example\_recurrenttx\_refund.php– shows an example of how to refund a transaction in a payment cycle.
- 9.) example\_wallet\_refund.php– shows an example of how to refund a wallet based transaction.
- 10.) example\_recurrenttx\_void.php– shows an example of how to cancel a recurrent payment cycle.
- 11.) example\_recurrenttx\_getcarddata.php – shows an example of how to retrieve the basic card details associated with a recurrent transaction.
- 12.) ResponseCodes\_TestAccountsAndCards.doc - this file contains the response codes and their meanings as well as test accounts. The test accounts are provided to simulate different scenarios giving rise to various response codes. Ensure that while testing you get the required code for each of the test accounts.

Before starting the integration process you need the following:

- A database that logs the entire details of transaction.
- The URL of the notification page to which customers would be directed upon payment completion.

- Optional – The URL of the page to which payment notifications would be sent in the case of recurrent transactions.

## Calling the CashEnvoy Payment Interface

Once set up, the first step is to call the CashEnvoy payment interface. To do this, certain parameters must be sent via http post to the CashEnvoy payment interface URL. Check the files below.

- `example_callcepayment_interface.php` (normal payment)
- `example_callcepayment_interface_rc.php` (recurrent transaction with defined frequency)
- `example_callcepayment_interface_rc2.php` (recurrent transaction with NO defined frequency)

Also, note that you should log the details of every transaction in your database before calling the payment interface. The following are the post variables required by the gateway before the payment interface can be rendered correctly.

- 1.) Your CashEnvoy merchant id (provided by CashEnvoy) - must be posted as variable `ce_merchantid`.
- 2.) A transaction reference between 6 and 20 characters containing no special characters - must be posted as variable `ce_transref`.
- 3.) A transaction amount in Naira which is not less than N50 - must be posted as variable `ce_amount`.
- 4.) A customer identification (email, username etc.) which is unique to the customer - must be posted as variable `ce_customerid`.
- 5.) A short description of the transaction - must be posted as variable `ce_memo`.
- 6.) A notification URL which is the absolute URL of the page to which the user will be redirected to upon payment completion. This page contains code to retrieve transaction response - must be posted as variable `ce_notifyurl`. Sample code for this page can be found in the **`example_requery_usage.php`** file.
- 7.) A target window level (options: self or parent). This specifies the window level the notification URL should be loaded into. This is useful for those integrating via iframes and modal windows - must be posted as variable `ce_window`.
- 8.) An hmac signature posted as variable `ce_signature`. Details of how to generate this signature can be found in the **`example_callcepayment_interface.php`**, **`example_callcepayment_interface_rc.php`** or **`example_callcepayment_interface_rc2.php`** file.

- 9.) Optional: A transaction frequency (for recurrent transactions) which can be none, weekly, fortnightly, monthly or yearly. This must be posted as variable `ce_recurrent`. If this variable is set to “none”, the merchant is expected to make subsequent payment requests as and when necessary with no limitations on the time between transactions in a payment cycle.
- 10.) Optional: A CashEnvoy payment cycle management mandate (for recurrent transactions) which can be Yes, No or empty. This must be posted as variable `ce_cyclemgt`. If `ce_recurrent` is set to “none”, then this variable should be left empty.
- 11.) Optional: An IPN URL (posted as variable `ce_ipnurl`) which is the absolute URL of the page you want payment notifications sent to for future scheduled payments (recurrent transactions) or one-off payments. This page contains code to receive payment status. Sample code for this page can be found in the **example\_ipn\_usage.php** file.

These variables must be posted to **<https://www.cashenvoy.com/sandbox2/?cmd=cepay>** for the payment interface to be rendered. If the payment interface is not rendered, an appropriate error message will be displayed.

## Retrieving Payment Response

The example file explains how to retrieve the status of transactions. Here, you will need to provide a transaction reference, your merchant id and a request signature to the API in order to receive the transaction status. Optionally, you can also provide a format (xml or json) in which the response should be returned.

The transaction status comprises of the transaction reference, transaction amount and the response code.

See the **example\_requery\_usage.php** file for full details. The sample code in this example or any variations you may want to develop would be used in the notification URL page.

### **Note (Preventing Man in the middle attacks):**

When a successful response (C00) is received for a transaction, ensure that you compare the amount returned by CashEnvoy to the value recorded in your database against that transaction reference.

Only provide service to the customer when these amounts are equal. If they are not equal, do not provide service to the customer. Contact [support@cashenvoy.com](mailto:support@cashenvoy.com) providing the transaction reference.

This is also one of the mandatory tests specified in the User Acceptance Test document.

Also note that a payment notification will be sent to the IPN URL if one was provided when the payment interface was called.

## Refund (Wallet Payments Only)

You can also initiate a refund by calling the wallet refund api as seen in `example_wallet_refund.php`.

## Broken Transactions

A broken transaction is one in which there is break in the internet connection before the customer can be provided with a page showing the transaction status. Merchants are required to always log transactions before calling the payment interface. In case of a broken transaction leading to a customer complaining that he has been billed but no service has been given, merchant should log into the CashEnvoy Panel to confirm the transaction status. Alternatively, the merchant may set up an independent requery page on its backend. The input would be the transaction reference which should then be used to obtain payment status as shown in the **example\_requery\_usage.php** file.

## Recurrent Transactions – Taking Future Payments

Once a recurrent transaction is created, future payments can be taken in 2 ways:

- 1) Automatically by CashEnvoy (in line with specified frequency – weekly, monthly etc.) **if the variable ce\_cyclemgt was set to Yes**. A notification is sent to the merchant IPN URL. Failed payments are retried 5 times (once per day).
- 2) Manually by calling the recurrent transaction payment api **if the variable ce\_cyclemgt was set to No or empty and**
  - If **ce\_recurrent** was set to none, there will be no restrictions on the time between transactions in a payment cycle. `example_recurrenttx_pay2.php` shows how to do this.
  - If **ce\_recurrent** was to a defined period (i.e. weekly, monthly etc.), the selected frequency will be enforced between transactions in a payment cycle. `example_recurrenttx_pay.php` shows how to do this.

## Recurrent Transactions – Refund

1.) Login to your CashEnvoy account and go to Payment -> Recurrent Transactions

This provides a list of recurrent transactions with the ability to refund a successful transaction if needed. Every refund will trigger a payment notification (of type “refund”) which is sent to the IPN URL specified when the payment cycle was created.

Note that if you did not specify the IPN URL when creating the recurrent transaction, you will not be able to refund a transaction via the CashEnvoy panel. You will have to use option 2 below.

2.) You can also initiate a refund by calling the recurrent transaction refund api as seen in `example_recurrenttx_refund.php`.

## Recurrent Transactions – Cancel

1.) Login to your CashEnvoy account and go to Payment -> Pre-Authorizations

This provides a list of pre-authorized transactions with the ability to cancel an active authorization if needed thereby halting all future payment cycles. Every cancellation will trigger a notification (of type “cancellation”) which is sent to the IPN URL specified when the payment cycle was created.

**Note that if you did not specify the IPN URL when creating the recurrent transaction, you will not be able to cancel it via the CashEnvoy panel. You will have to use option 2 below.**

2.) You can also cancel by calling the recurrent transaction cancellation api as seen in `example_recurrenttx_void.php`.

## Additional Information

The User Acceptance Test document provides the criteria that must be met before a merchant can go live. Please read and fulfil the requirements before requesting a go-live test. Also, note the service URLs for the test and live environments. Once approved to go live, the live environment variables should be used.

	Test Environment	Live Environment
Payment Interface	<a href="https://www.cashenvoy.com/sandbox2/?c_md=cepay">https://www.cashenvoy.com/sandbox2/?c_md=cepay</a>	<a href="https://www.cashenvoy.com/webservice/?c_md=cepay">https://www.cashenvoy.com/webservice/?c_md=cepay</a>
Retrieve Status	<a href="https://www.cashenvoy.com/sandbox2/?c_md=requery">https://www.cashenvoy.com/sandbox2/?c_md=requery</a>	<a href="https://www.cashenvoy.com/webservice/?c_md=requery">https://www.cashenvoy.com/webservice/?c_md=requery</a>
Refund (Wallet Payments)	<a href="https://www.cashenvoy.com/sandbox2/?c_md=ce_wrefund">https://www.cashenvoy.com/sandbox2/?c_md=ce_wrefund</a>	<a href="https://www.cashenvoy.com/webservice/?c_md=ce_wrefund">https://www.cashenvoy.com/webservice/?c_md=ce_wrefund</a>
Recurrent Transactions - Future Payment	<a href="https://www.cashenvoy.com/sandbox2/?c_md=ce_rcpay">https://www.cashenvoy.com/sandbox2/?c_md=ce_rcpay</a>	<a href="https://www.cashenvoy.com/webservice/?c_md=ce_rcpay">https://www.cashenvoy.com/webservice/?c_md=ce_rcpay</a>
Recurrent Transactions - Refund	<a href="https://www.cashenvoy.com/sandbox2/?c_md=ce_rcrefund">https://www.cashenvoy.com/sandbox2/?c_md=ce_rcrefund</a>	<a href="https://www.cashenvoy.com/webservice/?c_md=ce_rcrefund">https://www.cashenvoy.com/webservice/?c_md=ce_rcrefund</a>
Recurrent Transactions - Cancellation	<a href="https://www.cashenvoy.com/sandbox2/?c_md=ce_rcvoid">https://www.cashenvoy.com/sandbox2/?c_md=ce_rcvoid</a>	<a href="https://www.cashenvoy.com/webservice/?c_md=ce_rcvoid">https://www.cashenvoy.com/webservice/?c_md=ce_rcvoid</a>

## **Additional Notes**

The sandbox supports only payments via the CashEnvoy wallet, local and international cards for testing purposes but all other payment channels (GTBank, etransact) are available on the live environment. Once you are able to successfully integrate using the sandbox, no further development is required in live for the additional payment channels.