CCAvenue®

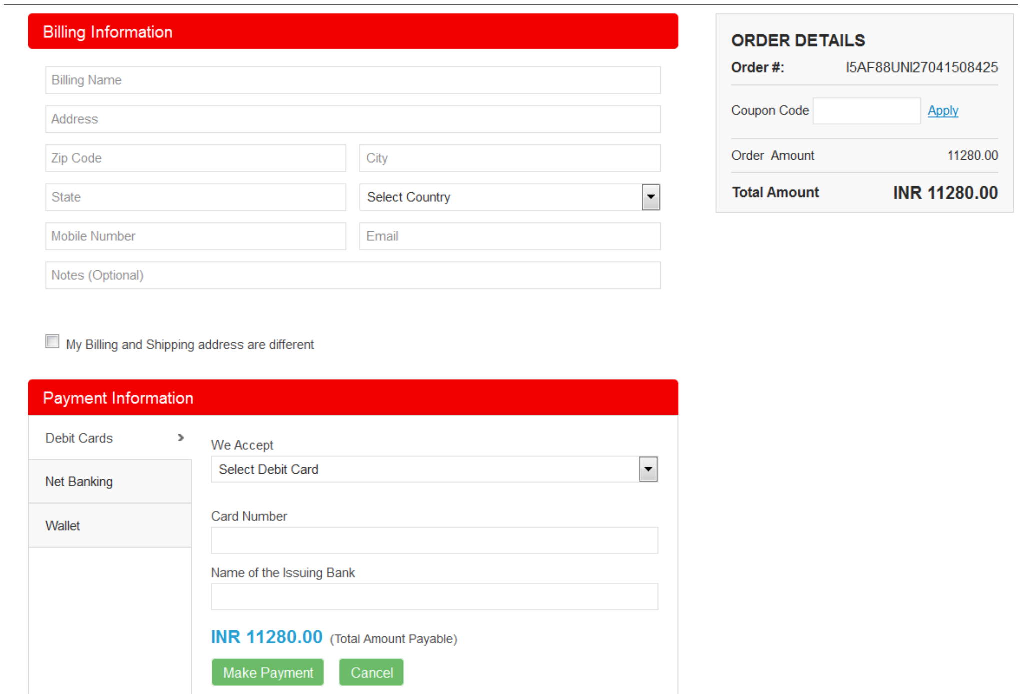
Knowledge Hub

VERSION 1.0

4/24/2015

TRANSACTION FLOW

1.1 CCAvenue billing page (Non-Seamless) - Avoid the hassle of developing and managing your own checkout page. Use the customizable billing page provided by CCAvenue which enables you to collect billing and shipping information of the customer.



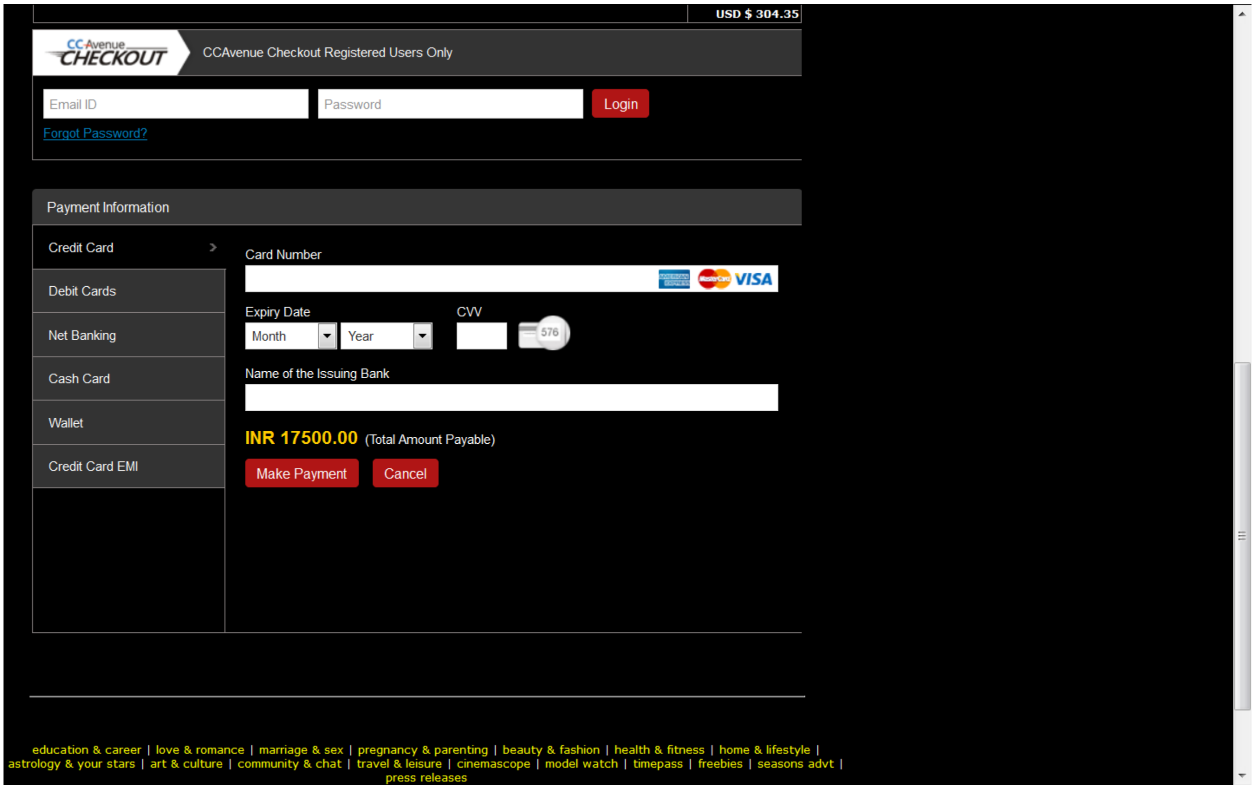
ORDER DETAILS

Order #:	I5AF88UNI27041508425
Coupon Code	Apply
Order Amount	11280.00
Total Amount	**INR 11280.00**

INR 11280.00 (Total Amount Payable)

1.2. CCAvenue iFrame checkout - Fastest and easiest way to enable payments on your website. CCAvenue iframe checkout is a pre-configured form, which validates the payment data, allows users to store card information to expedite the payment process in future. CCAvenue iFrame checkout also handles PCI compliance.

Collect card credentials and accept payments on your checkout page using our secure iFrame integration without being PCI DSS certified. Reduce payment hops and allow customers to make secure payments without leaving your webpage for a seamless brand experience.



USD $ 304.35

CCAvenue Checkout Registered Users Only

Email ID

Password

Login

Forgot Password?

Payment Information

Credit Card

Debit Cards

Net Banking

Cash Card

Wallet

Credit Card EMI

Card Number

Expiry Date

Month

Year

CVV

Name of the Issuing Bank

INR 17500.00 (Total Amount Payable)

Make Payment

Cancel

education & career | love & romance | marriage & sex | pregnancy & parenting | beauty & fashion | health & fitness | home & lifestyle | astrology & your stars | art & culture | community & chat | travel & leisure | cinemascope | model watch | timepass | freebies | seasons advt | press releases

1.3 a. Custom checkout form - Merchants can build a custom checkout form to collect order and payment information and pass on the same to CCAvenue server for payment processing. CCAvenue can also store the payment information of the customer to expedite the payment process in future.

1.3 b. Direct Connect - This integration enables you to deliver payment services directly through your website without redirecting your users to CCAvenue. This integration gives you the control to not only build your own custom checkout form, but also control the payment request process with the banks. On submission of the payment information, merchant initiates a server-to-server call to CCAvenue to fetch the request payload for the payment option selected by the user. The request payload received from CCAvenue will be used by you to connect directly to the bank's authentication/3D secure page, bypassing CCAvenue. However the response handling would be at CCavenue. CCAvenue will receive the authentication status from the bank and in turn post the transaction status back to the merchant's website.

1.4. CCAvenue shopping cart - CCAvenue provides merchants with a product management module and a customizable shopping cart, thereby eliminating the need for developing/maintaining their own. Integrate the CCAvenue Shopping Cart with your website and start selling online in less than 24 hours with the most basic HTML knowledge.

Simply add your products details, insert "Add to cart" button for all products and link it to the CCAvenue gateway. Customize the shopping cart page to match the look of your website and you're ready to sell online.

CCAvenue Shopping Cart is designed using Responsive Web Design approach and is built to adapt to all devices. This enables your customers to shop even on the go.

CONFIRMING A TRANSACTION

On successful completion of any transaction, the transaction needs to be confirmed within 12 days, to claim funds. If the transaction is not confirmed funds are released back to the customer's account.

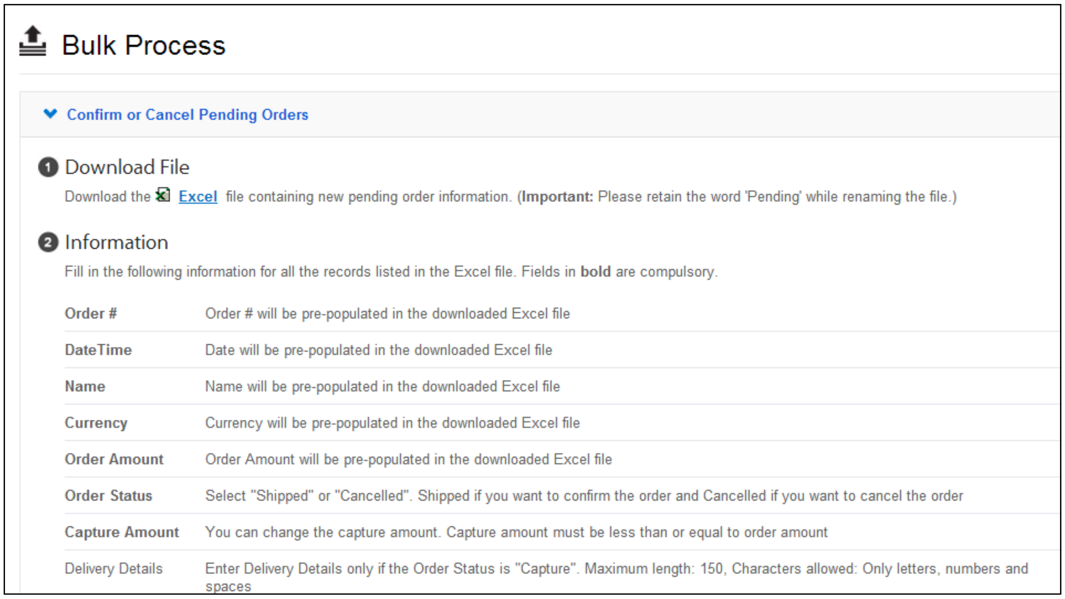
Please find various modes of confirmation:

Confirmation of Orders Manually

You need to confirm or cancel a pending order within 12 days after authorization.

1. These tasks can be executed from the Orders>Pending Orders page in your M.A.R.S. account, which displays a list of pending orders.
2. To confirm or cancel an order:
   (a) Click on the order number to view its entire details in a pop-up window.
   (b) Click on Confirm Or Cancel button on the pop-up window to confirm or cancel the order.
3. If you wish to confirm or cancel a batch of orders:
   (a) Select the check boxes of the corresponding orders displayed on the Pending Orders page
   (b) Click the appropriate button (Confirm All/Cancel All) at the bottom of the page.

Confirming Pending Orders in Bulk



Bulk Process

Confirm or Cancel Pending Orders

1 Download File

Download the Excel file containing new pending order information. (**Important:** Please retain the word 'Pending' while renaming the file.)

2 Information

Fill in the following information for all the records listed in the Excel file. Fields in **bold** are compulsory.

Order #	Order # will be pre-populated in the downloaded Excel file
DateTime	Date will be pre-populated in the downloaded Excel file
Name	Name will be pre-populated in the downloaded Excel file
Currency	Currency will be pre-populated in the downloaded Excel file
Order Amount	Order Amount will be pre-populated in the downloaded Excel file
Order Status	Select "Shipped" or "Cancelled". Shipped if you want to confirm the order and Cancelled if you want to cancel the order
Capture Amount	You can change the capture amount. Capture amount must be less than or equal to order amount
Delivery Details	Enter Delivery Details only if the Order Status is "Capture". Maximum length: 150, Characters allowed: Only letters, numbers and spaces

1. In your M.A.R.S account, click on the Orders > Bulk Process, to open the Bulk Process page.
2. Click the Confirm or Cancel Pending Orders section to expand the section and view the details.

3. Click on the Excel link to download the format for uploading you details.
4. Please retain the word "Pending" while you rename the excel file.
5. Edit the downloaded Excel file with the Pending order information.
6. Enter the relevant details in the fields provided in the Excel file and save the details. Mandatory fields include Order No, Currency and Order Amount.
7. Upload the Excel file with complete capture details to confirm or cancel the orders.
8. Maximum file size for Bulk uploads is 1 MB.

Confirm API call allows you to confirm a pending order. Only confirmed orders are settled into the merchant's account. An order older than 12 days is automatically cancelled. Once an order has been auto-cancelled by the system, it cannot be confirmed.

You can optionally confirm only part of an order. Remember, you can do so only once. Remaining funds will be refunded to the credit card or debit card or net banking account that was originally charged.

Request Parameters

Name	**Description**	**Note**
enc_request (*required*)	AES encrypted string contening request parameters.	
access_code (*required*)	Unique CCavenue access code which is generated when merchant registered their IP address. You must send this with each request.	
request_type (*required*)	API requests are accepted in XML, JSON or String. Specify the request type.	Possible value for request_type is "**XML**" or "**JSON**" or "**STRING**".
response_type (*optional*)	API returns responses in XML, JSON or String format. If left blank, the response will be in the same format as request.	Possible value for respose_type is is "**XML**" or "**JSON**" or "**STRING**".
Command (*required*)	Command value specify the API calls. You must send this with each request.	Possible value for this API call is "**confirmOrder**".
reference_no *(required)*	Unique CCAvenue reference number for the transaction.	Numeric(25) .
amount	Transaction amount to be captured. Amount can be full or partial of the transaction	Decimal(12,2).

(required)	amount.	

Response Parameters

Name	Description	Note
status	This states whether the call was successful or not. If value of this parameter is "1" then you need not decrypt the enc_response as it will contain plain error message.	Value "0" denotes the API call success. Value "1" denotes API call failure. On enc_response is plain text represents the error message.
enc_response	AES encrypted response containing format as per response_type	
success_count	Merchant checks the successfully processed records for confirmation of the transactions.	Numeric 0<=**success_count**<= Number of orders to be confirm.
reference_no	Unique CCAvenue reference numbers for the transaction.	Numeric(25)
reason	Failure reason if the given unique reference_no not going to confirm.	String please refer below table for failure message.

Note: You will have to decrypt the above response from "enc_response" parameter. Kindly refer to the decryption section.

Reason / Error_desc	**Applicable to**
"Order is already confirmed."	Confirm

“Invalid order status:”	Confirm,Cancel,
“Pending Risk Mitigation:”	Confirm,Cancel,
“Please provide at least one order list.”	Confirm,Cancel,
“Duplicate reference number value.”	Confirm,Cancel,

Auto confirm with Risk Mitigation:

This is configurable at CCAvenue, on enabling this feature the successful transactions are converted to shipped status post risk mitigation. These transactions will now be seen in unsettled section awaiting payout.

Auto confirm without Risk Mitigation:

This is configurable at CCAvenue, on enabling this feature the successful transactions are converted to shipped status in real-time and ***risk mitigation would not be taken care at CCAvenue***. These transactions will now be seen in unsettled section awaiting payout.

REFUND

Granting Full or Partial Refund Manually (Manual Refund)

You can grant full or partial refunds of shipped orders with our Order Lookup functionality.

1. In your M.A.R.S account, click on the Orders > Order Lookup, to open the Order Lookup pop up.

2. Choose the date range from the option available on the pop up window.

3. You need to select the ‘Shipped’ option from the list on the Order Lookup pop up and click the ‘Search’ button.

4. From the list of shipped orders, select the order(s) for which you wish to grant refund.

5. To view the details of the order against which you wish to grant refund, click the specific order number in the list of shipped orders. The order details are displayed in the pop-up.

6. Click the refund link in the pop-up to grant the refund.

Note: You can also directly click the refund link against the specified order displayed in the list of shipped orders to access this refund option.

7. Select the “Full Refund” or “Partial Refund” option in the Refund Payment pop-up.

8. Specify the refund amount if you have selected the partial refund amount.

Note: The amount cannot be greater than the amount of the order.

9. Click the Cancel button if you want to abort the refund process.

10. Click the Refund button to complete the refund process.

Important Info for Refund:

1. Refunds will be initiated for Confirmed orders and orders of the gateway which is set to Auto-capture/ Sale Mode, netbanking, debit card etc. These orders will create a refund entry.

2. Refunds shall be initiated only against Confirmed amount (not order amount).

3. Multiple Refunds for the same order can be performed if the same is enabled for the merchant (Please contact CCAvenue for the same). However, multiple refunds are solely depends on the gateway that allows it.

4. Refunds once initiated cannot be altered later by the merchant.

Refunding Orders in Bulk

1. In your M.A.R.S account, click on the Orders > Bulk Process, to open the Bulk Process page.

2. Click the Refund Orders section to expand the section and view the details.

3. Click on the Excel link to download the format for uploading your details.

4. Please retain the word "Refund" while you rename the excel file.

5. Edit the downloaded Excel file with the refund order information.

6. Enter the relevant details in the fields provided in the Excel file and save the details. Mandatory fields include Order No, Capture Amount, Refunded Amount and Refund Amount. **Note:** While the Refunded Amount field denotes the amount already refunded till date, the Refund Amount field denotes the amount yet to be refunded. The Refund Amount must be less than or equal to the Capture Amount.

7. Upload the Excel file with all the refund details to complete the refund transaction.

8. Maximum file size for Bulk uploads is 1 MB.

Important Info for Bulk Refund:

1. All columns are mandatory.
2. Serial number is exclusive within the order – refund cannot be initiated with same serial number for the same order more than once (for multiple refunds).
3. After filling the data, upload the file. Except invalid rows all other rows will get processed. The error messages for the error entries will be displayed on your screen.
4. Following is the table of various invalid scenarios and their corresponding error messages:

Scenario	**Error message**
If Refund file name doesn't include the word 'Refund'	Invalid file name. File name should contain the word Refund'
If Refund is blocked for the merchant	Refund is blocked. Please contact CCAvenue for more details.
When a wrong file/file format is uploaded	Please upload valid .XLS file.
When an empty file is uploaded	No Records updated
When order no/tracking id does not match or exists	Invalid tracking id/order no.
When Multiple refund is not allowed	Multiple refunds are not allowed

When the order is already fully refunded or Refund amount is greater than Confirmed amount	Either Order is fully refunded or Refund amount entered exceeds Capture amount.
When the order is not in a refundable status (apart from shipped/refunded)	Order not in refundable status

Refund API:

The Refund API call allows you to refund an order/transaction that has previously been executed but not yet refunded. Funds will be refunded to the credit card or debit card or net banking account that was originally charged.

Request Parameters

Name	**Description**	**Note**
enc_request (*required*)	AES encrypted request data	
access_code (*required*)	Unique CCavenue access code which is generated when merchant registered their IP address. You must send this with each request.	
request_type (*required*)	API requests are accepted in XML, JSON or String. Specify the request type.	Possible value for request_type is "**XML**" or "**JSON**" or "**STRING**".
response_type (*optional*)	API returns XML, JSON or String responses. If left blank, the response will be in the same format as request.	Possible value for respose_type is is "**XML**" or "**JSON**" or "**STRING**".
command (*required*)	Unique name which is specify the API Call. You must send this with each request	Possible value for command to this API call is "**refundOrder**".
reference_no *(required)*	Unique CCAvenue reference number for the transaction.	Numeric(25)
refund_amount *(required)*	Transaction amount to be refund. Amount can be full or partial of the transaction amount.	Decimal(12,2).

refund_ref_no *(required)*	Unique reference number shared by merchant to refund the captured transaction amount.	AlphaNumeric(30)

Response Parameters

Name	**Description**	**Note**
status	This states whether the call was successful or not. If value of this parameter is "1" then you need not decrypt the enc_response as it will contain plain error message.	Value "0" denotes the API call success. Value "1" denotes API call failure. On enc_response is plain text represents the error message.
enc_response	AES encrypted response containing format as per *response_type*	
refund_status	Refund status specify that given refund request is going to success or failure.	Numeric Value "0" denotes refund success Value "1" denotes refund failure.
reason	Failure reason if the refund request is going to failed for the transactions.	String please refer below table for failure message.

INSTANT GRATIFICATION (IG):

In the ecommerce environment when a transaction traverses from merchant site to payment gateway to bank page there are possibilities that the transaction is charged on the bank page but when redirecting back to CCAvenue it drops due to

a) Session time out
b) Connectivity issue
c) Customer closes browser on the bank page

To handle such cases CCAvenue reconciles with the banks:

- Using server to server call (if banks provide the same) or
- Using the recon sheets which is mailed to us the next working day by the bank.

If the customer is charged and transaction status was unknown to CCAvenue in real-time we handle it in 2 ways based on merchant settings:

a) Case 1: Convert the awaited transaction as successful. This is apt for brick and mortar merchants. Such merchants are configured as non-instant gratification merchants.

b) Case 2: Reverse the charged transaction back to customer if status of the transaction is not known in real time.

This reversal can be an instant one or a timed activity.

a) Instant reversal: This is apt for real time service rendering merchants. Such merchants are configured as instant gratification merchant and the minute we get the status from the bank, we mark such transactions for reversal.

b) Timed Reversal: in case, the merchant can fulfill the order even if the status of a transaction in known within an "X" amount of time. We can configure the no. of hours OR a fixed time through the day, after which if the status of the transaction is communicated to us from the bank and if the customer is charged, the system would mark the same for reversal. If the status is known prior to the "configured time" we would pass it on to the merchant as a successful transaction.

If you wish to opt for the timed or instant gratification mode for your account, do let us know via email at service@ccavenue.com and we shall configure your account for the same.

Alternatively, merchant may decide on the fly which transactions need to be considered for IG and which on non-IG. This can be done using the parameter instant_gratification.

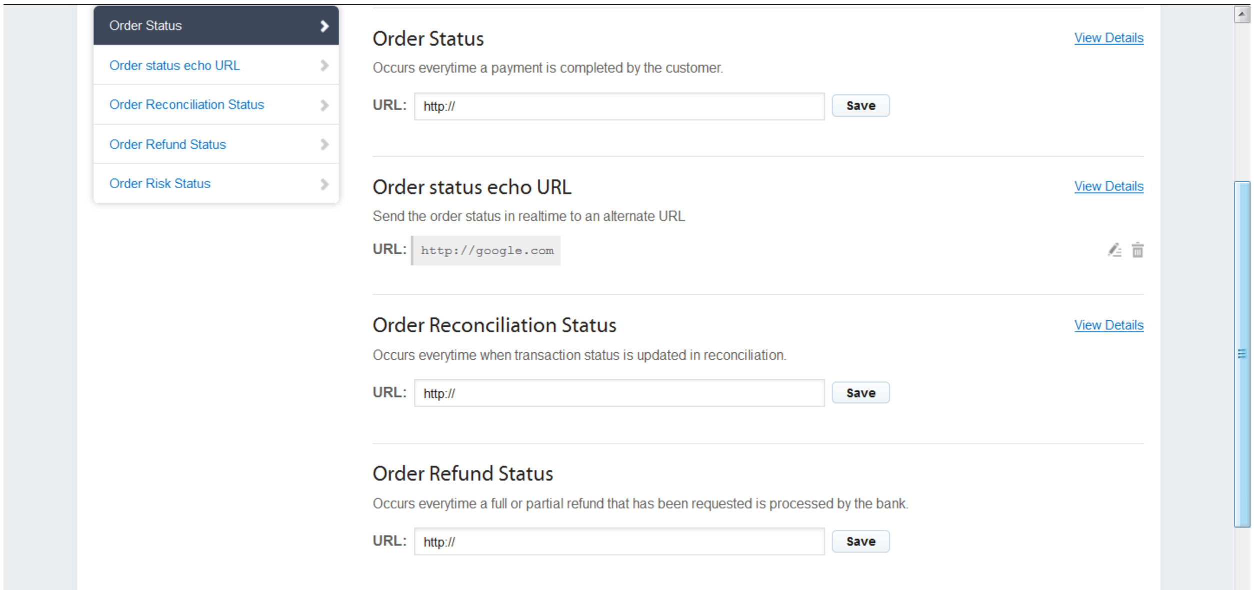
Example:

<input type="text" name="instant_gratification" value="Y"> (Transaction would be considered as IG)

<input type="text" name="instant_gratification" value="N"> (Transaction will not be considered as IG).

DYNAMIC EVENT NOTIFICATION

This feature will help you to fetch the transaction details through an asynchronous method. For every attempt CCAvenue will pass the "encresp" parameter to your configured URL. Kindly use the logic which you have applied for transaction value decryption. you will be able retrieve the response values which has been posted to you for the transaction. Register destination URLs that will receive notifications for order status, order refund status and order risk status that occur.



Order Status

Occurs everytime a payment is completed by the customer.

URL: http://

Order status echo URL

Send the order status in realtime to an alternate URL

URL: http://google.com

Order Reconciliation Status

Occurs everytime when transaction status is updated in reconciliation.

URL: http://

Order Refund Status

Occurs everytime a full or partial refund that has been requested is processed by the bank.

URL: http://

Configure URLs for Events

1. **Order Status:** If the merchant does not post a returnURL, and stores a URL in this section, the customer would be redirected to the page specified here. For eg in invoice payments the CCAvenue page loads with status of transaction, merchant may choose to display his page.

2. **Order status echo URL:** an asynchronous push of the response parameters is done in real time to this alternate URL.

3. **Order Reconciliation Status:** If the status of any transaction is updated in recon the same is posted to the merchant in the URL mentioned here.

4. **Order Refund Status:** If a full or partial refund has been issued and when our system accepts the refund the status is posted back to the merchant. You may want to update the same into your database.

CCAVENUE CHECKOUT

You may contact our support desk to enable this feature.

Comfortable and convenient checkout experience for repeat customers, CCAvenue Checkout offers a smooth, fast and frictionless buying experience. This feature enables customers to store their credit and debit card credentials, billing and shipping details in a PCI -compliant environment for future use.

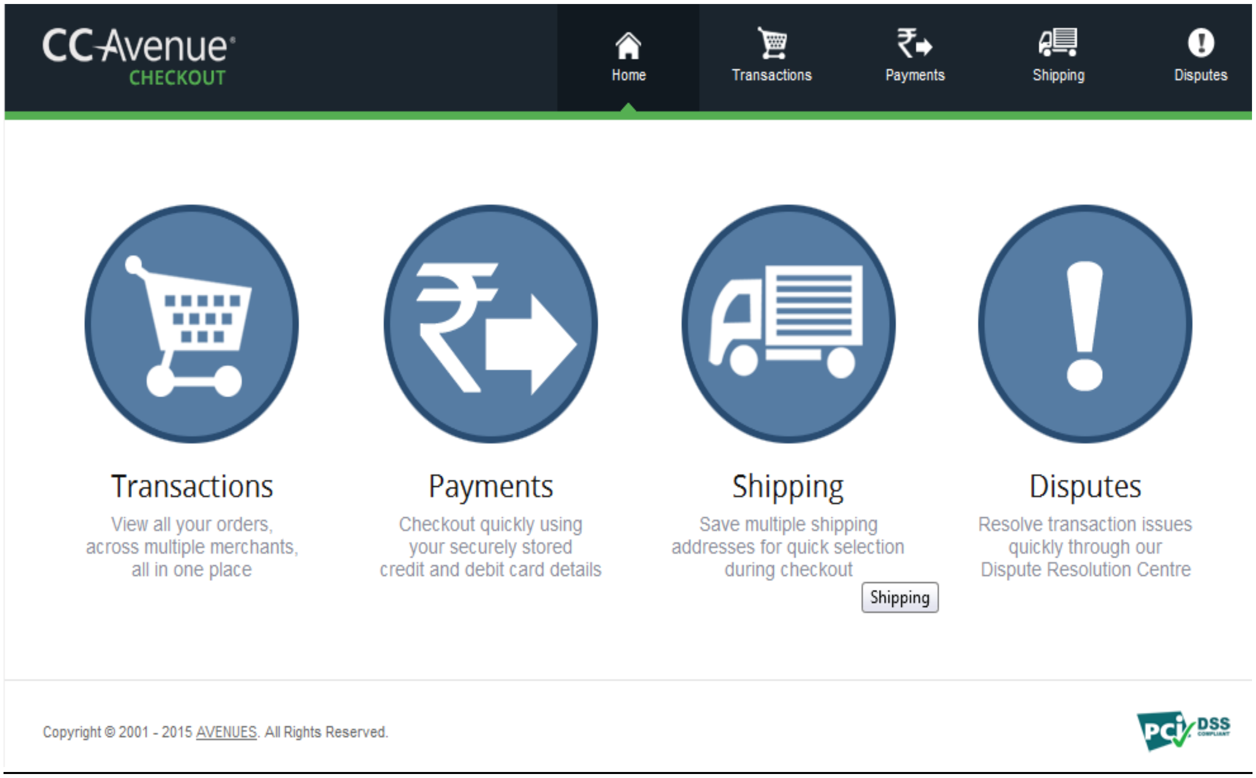
Card details are saved in an encrypted form and returning customers have to only enter the CVV and 3-D secure password when doing their next transaction. This facilitates speedy transactions, better customer retention, and encourages repeat purchases.

Managing the checkout account:

URL: https://www.ccavenue.com/buyers.jsp

CCAvenue provides the buyers with a panel to manage his checkout account. Here the buyer can:

a) View transactions performed by you performed across all the merchants.
b) Manage the card details stored securely on our PCI DSS servers.
c) Manage multiple shipping addresses for easy checkout process.
d) File a Dispute in case he is not happy with the services rendered by the merchant.



Transactions
View all your orders, across multiple merchants, all in one place

Payments
Checkout quickly using your securely stored credit and debit card details

Shipping
Save multiple shipping addresses for quick selection during checkout

Disputes
Resolve transaction issues quickly through our Dispute Resolution Centre

Copyright © 2001 - 2015 AVENUES. All Rights Reserved.

CCAVENUE VAULT

Vault Feature for Storing Card

CCAvenue enables the merchants to store card information of their customers for future transactions. This option is available in seamless and non-seamless implementations.

CCAvenue PG needs an additional parameter to identify your customer. You can send unique ID of the customer in your system at the time of initiating the transaction. This unique ID can be a customer ID, mobile number or an email ID. CCAvenue PG will store the card information against the customer identifier.

If there are any payment options stored against a customer identifier, CCAvenue PG will retrieve and load the same for customer to make the payment. Customer will also have an option of paying through a new card/payment option.

Vault Information

Name	**Description**	**Type (length)**
customer_identifier	The identifier against which the card information is to be stored or retrieved Email ID Customer ID Mobile number	Alphanumeric, '@' and '.' are allowed (70)

If vault data is stored then the merchant is posted a parameter vault=”Y” else vault=”N”.

Vault API: CCAvenue provides API calls to manage your vault, you can fetch the vault data, delete a customer or delete any payment option.

Delete Customer: The Delete Customer API call is used to delete the customer and all saved payment options for the customer.

Request Parameters:

Name	**Description**	**Note**
enc_request (*required*)	AES encrypted request data	
access_code (*required*)	This is the access code for your application. You must send this with each request.	
request_type	API requests are accepted in XML, JSON or String. Specify the request type.	Value is “XML” or “JSON” or “STRING”.

(*required*)		
response_type (*optional*)	API returns XML, JSON or String responses. If left blank, the response will be in the same format as request.	Value is "XML" or "JSON" or "STRING".
Command (*required*)	This is the command to access the API Calls. You must send this with each request.	Value is "**deleteCustomer**"
customer_id *(required)*	Customer ID to delete a particular customer.	Numeric

Response Parameters

Name	**Description**	**Note**
status	This states whether the call was successful or not. If value of this parameter is "1" then you need not decrypt the enc_response as it will contain plain error message.	Value "0" denotes the API call success. Value "1" denotes API call failure. On enc_response is plain text represents the error message.
enc_response	AES encrypted response containing format as per *response_type*	
deletion_status	Value 0 denotes successful deletion Value 1 denotes that customer could not be deleted.	Numeric Possible values for this is 0 - Deletion successful. 1 - Could not be deleted.
customer_id	Customer ID requested for deletion	Numeric
error_desc	This hold the reason if customer not deleted successfully.	String

Delete Customer Payment Option: The Delete Customer Payment Option API call is used to delete a particular payment option saved for the customer.

Request Parameters

Name	**Description**	**Note**
enc_request (*required*)	AES encrypted request data	
access_code (*required*)	This is the access code for your application. You must send this with each request.	
request_type (*required*)	API requests are accepted in XML, JSON or String. Specify the request type.	Value is "XML" or "JSON" or "STRING".
response_type (optional)	API returns XML, JSON or String responses. If left blank, the response will be in the same format as request.	Value is "XML" or "JSON" or "STRING".
command (*required*)	This is the command to access the API Calls. You must send this with each request.	Value is "**deleteCustomerPaymentOption**"
customer_id *(required)*	Customer ID against which the payment option is saved.	Numeric
customer_card_id *(required)*	Customer card ID is the unique identifier for the saved payment option.	Numeric

Response Parameters

Name	**Description**	**Note**
status	This states whether the call was successful or not. If value of this parameter is "1" then you need not decrypt the enc_response as it will contain plain error message.	Value "0" denotes the API call success. Value "1" denotes API call failure. On enc_response is plain text represents the error message.
enc_response	AES encrypted response containing format as per *response_type*	

customer_id	Customer ID against which the payment option was requested for deletion	Numeric
customer_card_id	lCustomer card ID to be deleted	Numeric
deletion_status	Status of deletion request	Possible values for this is 0 - Deletion successful. 1 - Could not be deleted.
error_desc	Reason for failure	String

Get Customer Payment Option: The Customer payment option API call is used to list payment option saved for a customer. Payments options are saved for a customer in vault for easy and convenient payment.

Request Parameters

Name	**Description**	**Note**
enc_request (*required*)	AES encrypted request data	
access_code (*required*)	This is the access code for your application. You must send this with each request.	
request_type (*required*)	API requests are accepted in XML, JSON or String. Specify the request type.	Value is "XML" or "JSON" or "STRING".
response_type (*optional*)	API returns XML, JSON or String responses. If left blank, the response will be in the same format as request.	Value is "XML" or "JSON" or "STRING".
Command (*required*)	This is the command to access the API Calls. You must send this with each request.	Value is "**getCustomerPaymentOptions**"
customer_id *(required)*	Customer ID against which payment options are saved.	Numeric

Response Parameters

Name	**Description**	**Note**
status	This states whether the call was successful or not. If value of this parameter is "1" then you need not decrypt the enc_response as it will contain plain error message.	Value "0" denotes the API call success. Value "1" denotes API call failure. On enc_response is plain text represents the error message.
enc_response	AES encrypted response containing format as per *response_type*	
pay_Opt_List	List of payment options against the	

	requested Customer ID	
customer_id	Customer ID against which the payment options are saved.	Numeric
customer_card_no	Last four digit of card no	Will contain values only for Credit card and Debit card
customer_payopt_type	Payment option type	String OPTNBK-net banking OPTCRD-credit card OPTDBRD-debit card OPTMOBP-mobile payment OPTIVRS- OPTWLT-
customer_card_name	Customer card name	Alphanumeric with special characters (space, underscore, hyphen, apostrophe)
customer_email	Customer email id	Alphanumeric with special characters (dot, underscore, at)
customer_card_label	Customer card label.	String
customer_card_id	Customer card ID is the identifier for the payment options against the Customer ID	Numeric
error_desc	Contains description of errors	String

You may contact our support desk to enable this feature.

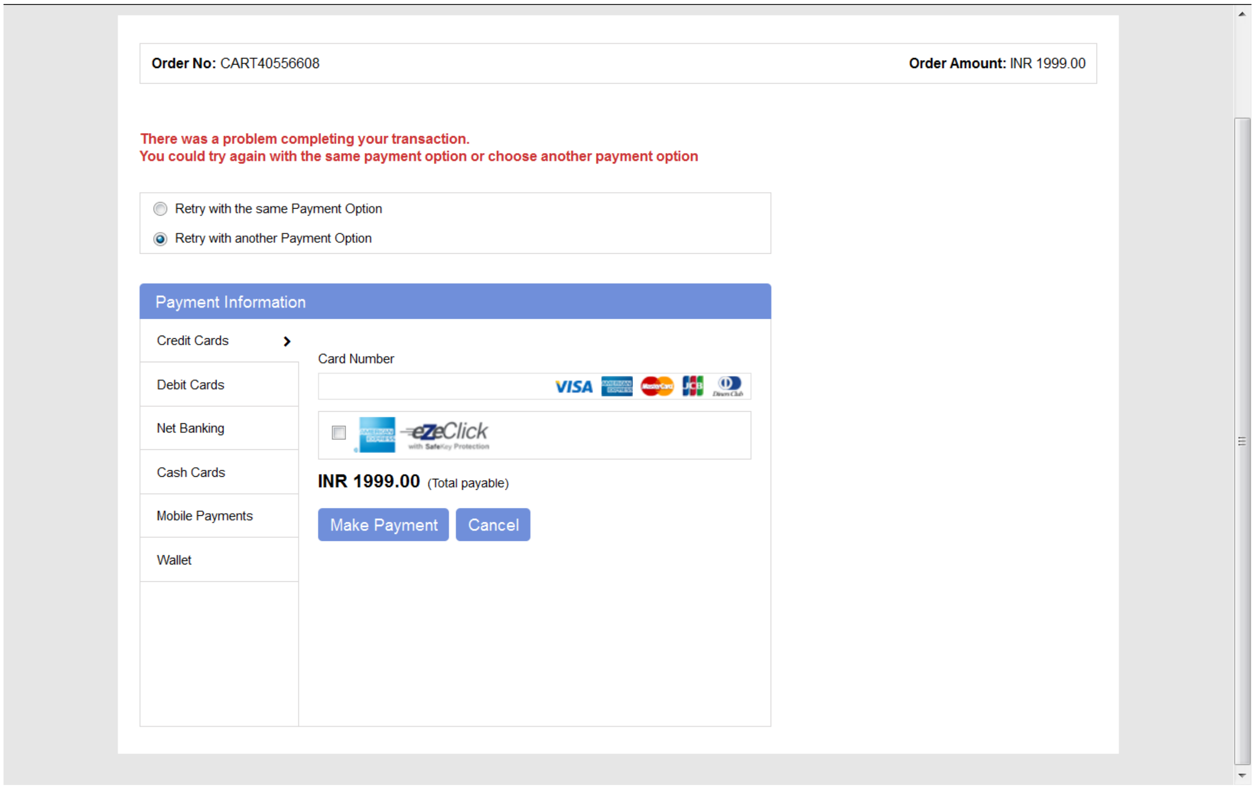
RETRY

CCAvenue offers a “Retry” facility to its customers that allows them to retry any transaction that fails due to reasons like wrong card details, incorrect payment option selected, bank downtime, etc.. In such case going through the entire process of filling the payment details can be tedious and time consuming.

The customer can retry using the same payment option or any of the other payment options assigned to the merchant. This ensures better success rate to the merchants and great convenience to the customers who do not have to return to merchant website to retry payment.

When the customer's transaction fails With our smart retry system, these steps can be avoided and instead the customer is redirected to an intermediary page showing the reason of the failure. He gets upto three attempts to use the same payment option or choose an alternate payment option to complete the purchase successfully. Retry option helps reduce number of abandoned trans-actions and boosts success rate more than 15%!

Currently, this feature is available on request basis. In case if you wish to opt for this feature kindly drop in an email at service@ccavenue.com and specify your Merchant id and Registered URL.



Order No: CART40556608 **Order Amount:** INR 1999.00

There was a problem completing your transaction.
You could try again with the same payment option or choose another payment option

○ Retry with the same Payment Option
● Retry with another Payment Option

Payment Information

Card Number

□ eZeClick with SafeKey Protection

INR 1999.00 (Total payable)

How does this work?

If any transaction fails the CCAvenue Retry page would allow the user to pay again using:

a) same payment option or
b) any other payment option assigned to the merchant.

The retry facility allows customers up to 3 more attempts to complete the transaction, which helps you close sales instantly and in **real-time** to augment the revenue stream of your business

Once the transaction is retried CCAvenue posts back the payment option which was used to complete the transaction. This information can be used to update your system. This will ensure there is no discrepancies in the 2 systems.

What happens if this retry page is closed?

In case this “Retry” page is closed by the customer, the transaction would be in awaited status in your system. But since this page appears only for failed transaction there will be no charge to the customer. You may use the status API to update your system on a scheduled basis for awaited transactions.

REPORTS

Reconciliation Report: By configuring this feature, the list of transactions processed through CCAvenue would be sent to you on a daily basis along with the order status as a dump. (Please note: Control to enable this feature is at our end, kindly confirm the list of email addresses to which the email has to be sent, so that we can configure.)

Settlement Report: In this report the transaction details would be sent to you for which payment released from CCAvenue. (Please note: Control to enable this feature is at our end, kindly confirm the list of email addresses to which the email has to be sent, so that we can configure.).

Apart from this the MARS provides an extensive reporting system, which allows you to view and download the following reports at your convenience.

1) Unsettled Transaction Summary: The Unsettled Transaction Summary provides details of your transactions which will be settled in the upcoming payout cycle.

2) Payout Summary: The Payout Summary provides details of your payout settlements.

3) Sales Report: The Sales Report provides period-wise details of your sales transactions.

4) Sales Report by Payment Type: The Sales Report by Payment Type provides period-wise details of your sales transactions according to the payment type. Thus indicating which Payment method is most popularly used on your portal.

5) Direct Bank Settlements : The Direct Bank Settlements Report provides details of your direct bank settlements which are done by the bank into your account.

6) CCAvenue Service Invoice: The CCAvenue Service Invoice provides details of your earnings from the CCAvenue Service Invoice transactions

7) Reconciliation Report : This report consists of records which are updated post our reconciliation with the banks.

If the status of the transaction was not updated in real time but the customer is charged, these transaction status is updated as successful for non IG merchants and reversed back to the customer for IG merchants. Such transactions can be extracted from these reports.

8) Aborted transaction report : This report consists of transactions which are cancelled on the CCAvenue page by the customer with reasons if specified by the customer.

9) Invalid request report : This report bears the list of transactions which is rejected by the PG as invalid parameters or restricted characters are posted to us. Please use this link : https://mars.ccavenue.com/mer_register/security_update.htm to see list of restricted characters.

10) Failure Analysis : This report consists of transactions which are Unsuccessful with appropriate failure reasons from the banks end. In case you observe a spike in failure ratio you may use this report for detailed analysis.

11) Refund Analysis : Gives a detailed analysis of Refunds initiated,Successfull refunds,Pending Refunds & Failed Refunds.

==

Failure reason for API call

Error_code	Short Description	Long Description	Reason / Error_desc	Applicable to
51001	Missing Parameter	Reference Number: Required parameter missing	“Ensure that the Reference Number parameter is not blank.”	Confirm, Cancel, Refund, Status, UpdateBillingDetails, UpdateMerchantParams, InvoiceLookUp,
51002	Invalid Parameter	Reference Number: Invalid Parameter	“Ensure that the Reference Number parameter is numeric.”	Confirm, Cancel, Refund, Status, OrderLookUp, UpdateBillingDetails, UpdateMerchantParams, InvoiceLookUp, GetPendingOrders
51003	Invalid Parameter	Reference Number: Invalid Parameter	“Ensure that the Reference Number parameter does not exceed 25 characters.”	Confirm, Cancel, Refund, Status, OrderLookUp, UpdateBillingDetails, UpdateMerchantParams, InvoiceLookUp,

				GetPendingOrders
51004	Invalid Parameter	Order Number: Invalid Parameter	“Ensure that reference no/order number is mandatory.”	Status
51006	Missing Parameter	Start/From Date: Required parameter missing	“Ensure that start/from date is provided.”	OrderLookUp
51007	Invalid Parameter	Start/From Date: Invalid Parameter	“Ensure that Start/From date is in dd-mm-yyyy format.”	OrderLookUp
51008	Invalid Parameter	End/To Date: Invalid Parameter	“Ensure that End/To date is in dd-mm-yyyy format.”	OrderLookUp
51009	Missing Parameter	Mobile number: Required parameter missing	“Ensure that Mobile Number parameter is not blank.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice, Order Lookup
51010	Invalid Parameter	Mobile number: Invalid Parameter	“Ensure that Mobile number parameter contains 10 digits.”	InvoiceLookUp, GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice, GetPendingOrders, OrderLookUp
51011	Missing Parameter	Email id: Required parameter missing	“Ensure that Email id is provided.”	GenerateQuickInvoice, GenerateRecurringInvoice
51012	Invalid Parameter	Email id: Invalid Parameter	“Ensure that only letters, numbers, hyphen, dot, one @ and underscore are provided for Email id parameter.”	InvoiceLookUp, GenerateQuickInvoice, GenerateRecurringInvoice, GetPendingOrders, Order Lookup
51013	Invalid Parameter	Email id: Invalid Parameter	“Ensure that Email id parameter does not exceed 70 characters.”	InvoiceLookUp, GenerateQuickInvoice, GenerateRecurringInvoice, GetPendingOrders
51014	Missing Parameter	Amount: Required parameter missing	“Ensure that Amount parameter is not blank.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice, Refund Order
51015	Invalid Parameter	Amount: Invalid Parameter	“Ensure that the Amount parameter is in Decimal.”	Confirm, Cancel, Refund, OrderLookUp, InvoiceLookUp,
51017	Invalid Parameter	Order Number: Invalid Parameter	“Ensure only letters, numbers, hyphen and underscore are provided for Order Number parameter.”	Status, GetPendingOrders, OrderLookUp

51018	Invalid Parameter	Order Number: Invalid Parameter	“Ensure that Order Number parameter does not exceed 30 characters.”	Status, GetPendingOrders, OrderLookUp
51020	Invalid Parameter	Minimum amount: Invalid Parameter	"Ensure that Minimum amount parameter has Decimal value."	Order Lookup
51022	Invalid Parameter	Maximum amount: Invalid Parameter	"Ensure that Maximum amount parameter has Decimal value."	Order Lookup
51024	Invalid Parameter	Page Number: Invalid Parameter	"Ensure that Page Number parameter is greater than zero."	Order Lookup
51026	Invalid Parameter	Invoice id: Invalid Parameter	“Ensure that invoice id parameter is numeric.”	Invoice Lookup
51028	Invalid Parameter	Invoice Number: Invalid Parameter	"Ensure that only letters, number hyphen and underscore are provided for invoice Number parameter."	Order Lookup
51029	Invalid Parameter	Invoice Number: Invalid Parameter	"Ensure that Invoice number parameter does not exceed 30 characters."	Invoice Lookup
51031	Invalid Parameter	Invoice type: Invalid Parameter	“Ensure that values for invoice type parameter are TASK/ITEM.”	InvoiceLookUp,
51032	Missing Parameter	Currency: Required parameter missing	“Ensure that Currency parameter is not blank.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice
51033	Invalid Parameter	Currency: Invalid Parameter	“Ensure that Currency is assigned to the merchant.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice
51034	Missing Parameter	Valid for: Required parameter missing	“Ensure that 'valid for' parameter is not blank.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice
51035	Invalid Parameter	Valid for: Invalid Parameter	“Ensure that 'valid for' parameter is Numeric.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice
51036	Invalid Parameter	Valid for: Invalid Parameter	“Ensure that 'valid for' parameter does not exceed 4 digits.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice
51037	Missing Parameter	Valid type: Required parameter missing	“Ensure that valid type parameter value is not blank.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice

51038	Invalid Parameter	Valid type: Invalid Parameter	“Ensure that values for valid type parameter are days/hours/minutes.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice
51039	Missing Parameter	Bill delivery type: Required parameter missing	“Ensure that bill delivery Type parameter is not blank.”	GenerateQuickInvoice
51040	Invalid Parameter	Bill delivery type: Invalid Parameter	“Ensure that value for bill delivery type parameter is SMS/EMAIL.”	GenerateQuickInvoice
51041	Missing Parameter	Name: Required parameter missing	“Ensure that Name parameter is not blank.”	UpdateBillingDetails, GenerateQuickInvoice, GenerateRecurringInvoice
51042	Invalid Parameter	Name: Invalid Parameter	“Ensure that only letters, space, dot, underscore, hyphen and single apostrophe are provided for Name parameter.”	UpdateBillingDetails, GenerateQuickInvoice, GenerateRecurringInvoice, PendingOrder
51043	Invalid Parameter	Name: Invalid Parameter	“Ensure that Name parameter does not exceed 60 characters.”	UpdateBillingDetails, GenerateQuickInvoice, GenerateRecurringInvoice, PendingOrder
51044	Missing Parameter	Address: Required parameter missing	“Ensure that Address parameter is not blank.”	UpdateBillingDetails
51045	Invalid Parameter	Address: Invalid Parameter	“Ensure only letters, numbers, space, hyphen, comma, ampersand, hash, circular brackets and dot are provided for Address parameter.”	UpdateBillingDetails
51046	Invalid Parameter	Address: Invalid Parameter	“Ensure that Address parameter does not exceed 315 characters.”	UpdateBillingDetails
51047	Missing Parameter	City: Required parameter missing	“Ensure that City parameter is not blank.”	Missing Mer_charge_reference_number, UpdateBillingDetails
51048	Invalid Parameter	City: Invalid Parameter	“Ensure that only numbers, letters, space, comma, hyphen and dot are provided for City parameter.”	UpdateBillingDetails
51049	Invalid Parameter	City: Invalid Parameter	“Ensure that City parameter does not exceed 30 characters.”	UpdateBillingDetails
51050	Missing Parameter	Zip/PIN code: Required parameter missing	“Ensure that Zip/PIN code parameter is not blank.”	UpdateBillingDetails
51051	Invalid Parameter	Zip/PIN code: Invalid Parameter	“Ensure that only letters, numbers, hyphen and space are provided for	UpdateBillingDetails

			Zip/PIN code parameter.”	
51052	Invalid Parameter	Zip/PIN code: Invalid Parameter	“Ensure that Zip/PIN code parameter does not exceed 15 characters.”	UpdateBillingDetails
51053	Missing Parameter	State: Required parameter missing	“Ensure that State parameter is not blank.”	UpdateBillingDetails
51054	Invalid Parameter	State: Invalid Parameter	“Ensure that only letters, hyphen, dot and space are provided for State parameter.”	UpdateBillingDetails
51055	Invalid Parameter	State: Invalid Parameter	“Ensure that State parameter does not exceed 30 characters.”	UpdateBillingDetails
51056	Missing Parameter	Country : Required parameter missing	“Ensure that Country Code parameter is not blank.”	UpdateBillingDetails
51057	Invalid Parameter	Country: Invalid Parameter	“Ensure that only letters and space are allowed for Country Code parameter.”	UpdateBillingDetails
51058	Invalid Parameter	Country: Invalid Parameter	“Ensure that Country Code parameter does not exceed 30 characters.”	UpdateBillingDetails
51059	Missing Parameter	Customer id: Required parameter missing	“Ensure that Customer id parameter is not blank.”	GetCustomerPaymentOptions, DeleteCustomerPaymentOption, DeleteCustomer, Customer payment option
51060	Invalid Parameter	Customer id: Invalid Parameter	“Ensure that only numerical values are provided for Customer id parameter.”	GetCustomerPaymentOptions, DeleteCustomerPaymentOption, DeleteCustomer
51061	Invalid Parameter	Customer id: Invalid Parameter	“Ensure that Customer id parameter does not exceed 25 digits.”	GetCustomerPaymentOptions, DeleteCustomerPaymentOption, DeleteCustomer, Customer payment option
51062	Missing Parameter	Customer card id: Required parameter missing	“Ensure that Customer card id is not blank.”	DeleteCustomerPaymentOption
51063	Invalid Parameter	Customer card id: Invalid Parameter	“Ensure that only numbers are provided for Customer card id parameter.”	DeleteCustomerPaymentOption

51064	Invalid Parameter	Customer card id: Invalid Parameter	“Ensure that Customer card id parameter does not exceed 25 characters.”	DeleteCustomerPaymentOption
51065	Missing Parameter	Parameter value: Required parameter missing	“Ensure that Merchant Params value is not blank.”	UpdateMerchantParams
51066	Invalid Parameter	Parameter value: Invalid Parameter	“Ensure that Alphanumeric, comma, hyphen, backslash and dot are provided for parameters Value.”	UpdateMerchantParams
51067	Invalid Parameter	Parameter value: Invalid Parameter	“Ensure that Parameter value does not exceed 100 characters.”	UpdateMerchantParams
51068	Missing Parameter	Bill id: Required parameter missing	“Ensure that Bill id parameter is not blank.”	UpdateInvoiceReferenceNumber, Updatemerchantreferencenumber
51069	Invalid Parameter	Bill id: Invalid Parameter	“Ensure that Bill id parameter is a numeric value greater than zero.”	UpdateInvoiceReferenceNumber, InvoiceLookUp, Updatemerchantreferencenumber
51070	Invalid Parameter	Bill id: Invalid Parameter	“Ensure that Bill id parameter does not exceed 25 characters.”	UpdateInvoiceReferenceNumber, InvoiceLookUp
51071	Missing Parameter	Merchant reference number: Required parameter missing	“Ensure that merchant Reference Number is not blank.”	UpdateInvoiceReferenceNumber,
51072	Invalid Parameter	Merchant reference number: Invalid Parameter	“Ensure that only letters, numbers, hyphen and underscore are provided for merchant Reference number parameter.”	UpdateInvoiceReferenceNumber, GenerateRecurringInvoice, GenerateInvoice
51073	Invalid Parameter	Merchant reference number: Invalid Parameter	“Ensure that Merchant reference number parameter does not exceed 25 characters.”	UpdateInvoiceReferenceNumber, GenerateRecurringInvoice, GenerateInvoice
51074	Missing Parameter	SMS Content: Required parameter missing	“Ensure that SMS Content parameter is not blank.”	GenerateRecurringInvoice, GenerateInvoice
51075	Missing Parameter	SMS Content: Required parameter missing	“Ensure that 'Basic place holders in SMS Content' parameter is not blank.”	GenerateRecurringInvoice, GenerateInvoice
51076	Invalid Parameter	SMS Content: Invalid Parameter	“Ensure that SMS content parameter does not exceed 500 Characters.”	GenerateRecurringInvoice, GenerateInvoice
51077	Missing Parameter	Email subject: Required parameter	“Ensure that Email Subject	GenerateQuickInvoice, GenerateRecurringInvoice,

		missing	parameter is not blank.”	GenerateInvoice
05107C	Invalid Parameter	Email subject: Invalid Parameter	“Ensure that Email subject parameter contains only letters, numbers, hyphens, dot, space and underscores.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice
51079	Invalid Parameter	Email subject: Invalid Parameter	“Ensure that Email subject parameter does not exceed 100 characters.”	GenerateQuickInvoice, GenerateRecurringInvoice, GenerateInvoice
51080	Invalid Parameter	Email Description: Invalid Parameter	“Ensure that Email Description parameter contains only letters, numbers, hyphen, dot, circular brackets, space, comma, underscore, hash and ampersand.”	GenerateRecurringInvoice, GenerateInvoice
51081	Invalid Parameter	Email Description: Invalid Parameter	"Ensure that Email Description parameter does not exceed 100 characters."	Generate QuickInvoice
51082	Invalid Parameter	File List: Invalid Parameter	“Ensure that only one file is attached in quick invoice.”	GenerateQuickInvoice
51083	Invalid Parameter	File List: Invalid Parameter	“Ensure that Attached File extension is (doc, jpg, jpeg, docx, pdf, png).”	GenerateQuickInvoice
51084	Invalid Parameter	File List: Invalid Parameter	“Ensure that File size does not exceed 1MB.”	GenerateQuickInvoice
51085	Missing Parameter	Refund Reference Number: Required parameter missing	“Ensure that 'refund reference no' parameter is not blank.”	Refund
51086	Invalid Parameter	Refund Reference Number: Invalid Parameter	“Ensure that Only letters and numbers are provided for refund Reference Number parameter.”	Refund
51087	Invalid Parameter	Refund reference number: Invalid Parameter	"Ensure that Refund reference number parameter does not exceed 100 characters."	Refund orders
51088	Missing Parameter	Frequency: Required parameter missing	“Ensure that Frequency parameter is not blank.”	GenerateRecurringInvoice
51089	Invalid Parameter	Frequency : Invalid Parameter	“Ensure that values for frequency parameter are only Daily/Monthly/Quarterly/Yearly.”	GenerateRecurringInvoice
51118	Missing Parameter	Occurrences: Required parameter missing	“Ensure that Occurrences parameter is not blank.”	GenerateRecurringInvoice

51119	Invalid Parameter	Occurrences: Invalid Parameter	“Ensure that Occurrences parameter is Numeric.”	GenerateRecurringInvoice
51120	Invalid Parameter	Occurrences: Invalid Parameter	“Ensure that Occurrences parameter does not exceed 3 digits.”	GenerateRecurringInvoice
51121	Invalid Parameter	Occurrences: Invalid Parameter	“Adding occurrences is only applicable for recurring invoice.”	GenerateRecurringInvoice
51122	Missing Parameter	Task name: Required parameter missing	“Ensure that task name parameter is not blank.”	GenerateRecurringInvoice, GenerateInvoice
51123	Invalid Parameter	Task name: Invalid Parameter	“Ensure that only letters, space, underscore and hyphen are provided for task name parameter.”	GenerateRecurringInvoice, GenerateInvoice
51124	Invalid Parameter	Task name: Invalid Parameter	“Ensure that Task name parameter does not exceed 30 Characters.”	GenerateRecurringInvoice, GenerateInvoice
51125	Missing Parameter	Task Note: Required parameter missing	"Ensure that Task Note parameter is not blank."	Generate Invoice
51126	Invalid Parameter	Task note: Invalid Parameter	“Ensure that Task note parameter contains only letters, numbers, hyphen, dot, circular brackets, space, comma, underscore, hash and ampersand.”	GenerateRecurringInvoice, GenerateInvoice
51127	Invalid Parameter	Task Note: Invalid Parameter	"Ensure that Task Note parameter does not exceed 60 characters."	Generate Invoice
51128	Missing Parameter	Task rate: Required parameter missing	“Ensure that task rate parameter is not blank.”	GenerateRecurringInvoice, GenerateInvoice
51129	Invalid Parameter	Task rate: Invalid Parameter	“Ensure that task rate parameter is decimal.”	GenerateRecurringInvoice, GenerateInvoice
51130	Missing Parameter	Task hour: Required parameter missing	“Ensure that task hour parameter is not blank.”	GenerateRecurringInvoice, GenerateInvoice
51131	Invalid Parameter	Task hour: Invalid Parameter	“Ensure that task hour parameter is numeric.”	GenerateRecurringInvoice, GenerateInvoice
51132	Invalid Parameter	Task hour: Invalid Parameter	“Ensure that task hour value does not exceed 4 digits.”	GenerateRecurringInvoice, GenerateInvoice
51133	Missing Parameter	Item name: Required parameter missing	“Ensure that item name parameter is not blank.”	GenerateRecurringInvoice, GenerateInvoice
51134	Invalid Parameter	Item name: Invalid Parameter	“Ensure that only letters, space,	GenerateRecurringInvoice,

			underscore and hyphen are provided for item name parameter.”	GenerateInvoice
51135	Invalid Parameter	Item name: Invalid Parameter	“Ensure that Item name parameter does not exceed 30 characters.”	GenerateRecurringInvoice, GenerateInvoice
51136	Missing Parameter	Item description: Required parameter missing	“Ensure that item description parameter is not blank.”	GenerateRecurringInvoice, GenerateInvoice
51137	Invalid Parameter	Item description: Invalid Parameter	“Ensure that Item description parameter contains only letters, numbers, hyphen, dot, circular brackets, space, comma, underscore, hash and ampersand.”	GenerateRecurringInvoice, GenerateInvoice
51138	Invalid Parameter	Item description: Invalid Parameter	“Ensure that Item description parameter does not exceed 60 characters.”	GenerateRecurringInvoice, GenerateInvoice
51139	Missing Parameter	Item Quantity: Required parameter missing	“Ensure that the item quantity parameter is not blank.”	GenerateRecurringInvoice, GenerateInvoice
51140	Invalid Parameter	Item Quantity: Invalid Parameter	“Ensure that the Item Quantity parameter is numeric.”	GenerateRecurringInvoice, GenerateInvoice
51141	Invalid Parameter	Item Quantity: Invalid Parameter	“Ensure that the Item Quantity parameter does not exceed 3 digits.”	GenerateRecurringInvoice, GenerateInvoice
51142	Missing Parameter	Unit cost: Required parameter missing	“Ensure that unit cost of item is not blank.”	GenerateRecurringInvoice, GenerateInvoice
51143	Invalid Parameter	Unit cost: Invalid Parameter	“Ensure that the Unit cost of item parameter is a Decimal Number.”	GenerateRecurringInvoice, GenerateInvoice
51144	Invalid Parameter	Terms and Conditions: Invalid Parameter	“Ensure that Only letters, numbers, hyphen, dot, circular brackets and ampersand are provided for Terms and Condition parameter.”	GenerateRecurringInvoice, GenerateInvoice
51145	Invalid Parameter	Terms and Conditions: Invalid Parameter	“Ensure that Terms and Conditions parameter does not exceed 500 characters.”	GenerateRecurringInvoice, GenerateInvoice
51146	Invalid Parameter	Due date: Invalid Parameter	“Ensure that Due date parameter is Numeric.”	GenerateRecurringInvoice, GenerateInvoice
51147	Invalid Parameter	Due date: Invalid Parameter	“Ensure that Due date parameter does not exceed 3 digits”	GenerateRecurringInvoice, GenerateInvoice
51148	Invalid Parameter	Due Date: Invalid Parameter	“Ensure that due date parameter is provided when late payment fees is	GenerateRecurringInvoice, GenerateInvoice

			present.”	
51149	Invalid Parameter	Late payment fees: Invalid Parameter	“Ensure that late payment fees parameter is provided when due date is present.”	GenerateRecurringInvoice, GenerateInvoice
51150	Invalid Parameter	Discount_if_paid_within_due_date: Invalid Parameter	“Ensure that value for discount_if_paid_within_due_date parameter is not greater than due date.”	GenerateRecurringInvoice, GenerateInvoice
51151	Missing Parameter	Discount Type: Required parameter missing	"Ensure that discount type parameter is provided when discount value is present."	GenerateRecurringInvoice, GenerateInvoice
51152	Missing Parameter	Discount value: Required parameter missing	"Ensure that discount value parameter is provided when discount type is present."	GenerateRecurringInvoice, GenerateInvoice
51153	Missing Parameter	Late payment fees type: Required parameter missing	"Ensure that late payment fees type parameter is provided when late payment fees is present."	GenerateRecurringInvoice, GenerateInvoice
51154	Missing Parameter	Late payment fees: Required parameter missing	"Ensure that late payment fees parameter is provided when late payment fees type is present."	
51155	Invalid Parameter	Discount value: Invalid Parameter	"Ensure that Discount value parameter ranges are: For percentage (1.00 -99.99) For flat (1.00-99999.99)"	GenerateRecurringInvoice, GenerateInvoice
51156	Invalid Parameter	Late payment fees: Invalid Parameter	"Ensure that Late payment fees parameter ranges are: For percentage (1.00 -99.99) For flat (1.00-99999.99)"	GenerateRecurringInvoice, GenerateInvoice
51157	Invalid Parameter	Tax List:Invalid Parameter	“Ensure that there are not more than two taxes for one task.”	GenerateRecurringInvoice, GenerateInvoice
51158	Invalid Parameter	Item/Task List: Invalid Parameter	“Ensure that at least one task/item is added.”	GenerateRecurringInvoice, GenerateInvoice
51159	Invalid Parameter	Task List: Invalid Parameter	“Ensure that the same task is not added twice.”	GenerateRecurringInvoice, GenerateInvoice

51160	Invalid Parameter	Item List: Invalid Parameter	“Ensure that the same item is not added twice.”	GenerateRecurringInvoice, GenerateInvoice
51161	Invalid Parameter	Discount value: Invalid Parameter	“Ensure that the Discount value parameter is Decimal.”	GenerateRecurringInvoice, GenerateInvoice
51162	Invalid Parameter	Late payment fees: Invalid Parameter	“Ensure that Late payment fees parameter is Decimal.”	GenerateRecurringInvoice, GenerateInvoice
51163	Invalid Parameter	Discount type: Invalid Parameter	“Ensure that values for Discount type parameter are Perc/Flat.”	GenerateRecurringInvoice, GenerateInvoice
51164	Invalid Parameter	Late payment fees type: Invalid Parameter	“Ensure that values for Late payment fees type parameter are Perc/Flat.”	GenerateRecurringInvoice, GenerateInvoice
51165	Invalid Parameter	Discount if paid within due date: Invalid Parameter	“Ensure that Discount if paid within due date parameter is Numeric.”	GenerateRecurringInvoice, GenerateInvoice
51166	Invalid Parameter	Discount if paid within due date: Invalid Parameter	“Ensure that Discount if paid within due date parameter does not exceed 3 digits.”	GenerateRecurringInvoice, GenerateInvoice
51167	Invalid Parameter	Tax List: Invalid Parameter	“Ensure that the same tax is not added twice for the same task/item.”	GenerateRecurringInvoice, GenerateInvoice
51168	Invalid Parameter	Tax List: Invalid Parameter	"Ensure that there are not more than two taxes applicable for one item."	Generate Invoice
51169	Invalid Parameter	Due Date: Invalid Parameter	"Ensure that Due date cannot be greater than valid for parameter."	Generate Recurring Invoice
51205	Invalid Parameter	Order List: Invalid Parameter	“Please provide at least one order list.”	Confirm, Cancel,
51206	Invalid Parameter	Order List: Invalid Parameter	“Ensure that Reference Number parameter is not repeated.”	Confirm, Cancel,
51207	Invalid Parameter	Start Date: Invalid Parameter	“Ensure that Start date is greater than or equal to current date.”	GenerateRecurringInvoice
51208	Invalid Parameter	Start Date: Invalid Parameter	“Adding start date is only applicable for recurring invoice.”	GenerateRecurringInvoice
51209	Invalid Parameter	Frequency: Invalid Parameter	“Adding frequency is only applicable for recurring invoice.”	GenerateRecurringInvoice, GenerateInvoice
51301	Invalid Parameter	Order List: Invalid Parameter	"'Order is already confirmed:'	Confirm Tracking Id is dynamically

			+{Tracking Id} + '#.' ”	appended
51302	Invalid Parameter	Order List: Invalid Parameter	"'Order is already Cancelled:'+{Tracking Id} + '#.' ”	Cancel Tracking Id is dynamically appended
51303	Invalid Parameter	Order List: Invalid Parameter	“'Invalid order status:' + {Tracking Id} + '#.' ”	Confirm, Cancel, Tracking Id is dynamically appended
51304	Invalid Parameter	Order List: Invalid Parameter	“'Invalid order/tracking id :' + {Tracking Id} + '#.' ”	Confirm, Cancel, Tracking Id is dynamically appended
51305	Invalid Parameter	Order List: Invalid Parameter	“ 'Pending Risk Mitigation:' + {Tracking Id} + '#.' ”	Confirm, Cancel, Tracking Id is dynamically appended
51306	Invalid Parameter	Order List: Invalid Parameter	“'Order Amt Mismatch:' + {Tracking Id} + '#.' ”	Confirm, Cancel, Tracking Id is dynamically appended
51307	Invalid Parameter	Order List: Invalid Parameter	“Pending risk mitigation.”	Confirm, Cancel
51308	Invalid Parameter	Order List: Invalid Parameter	“No record was found for the given criteria.”	Refund, Status, OrderLookUp,
51309	Invalid Parameter	Reference number: Invalid Parameter	“Ensure that merchant reference number parameter is not repeated.”	Refund
51310	Invalid Parameter	Reference number: Invalid Parameter	“Refund not allowed.”	Refund
51311	Invalid Parameter	Reference number: Invalid Parameter	“Multiple refunds are not allowed for the gateway.”	Refund
51312	Invalid Parameter	Reference number: Invalid Parameter	"Cannot initiate refund request for the given order."	Refund
51313	Invalid Parameter	Order List: Invalid Parameter	“No records found.”	Order Status, OrderLookUp,
51314	Invalid Parameter	Reference number: Invalid Parameter	“Refund limit exceeds.”	Refund
51315	Invalid Parameter	Reference number: Invalid Parameter	“This is a disputed order. You can refund only up to {Currency}{Refund	Refund, Currency and Refund amount

			amount limit}.”	limit is dynamically appended
51316	Invalid Parameter	Reference number: Invalid Parameter	“Ensure that Reg Id or Reference No parameters are not invalid.”	UpdateBillingDetails,
51317	Invalid Parameter	Reference number: Invalid Parameter	“Record already updated or Invalid Order Status.”	UpdateBillingDetails
51318	Invalid Parameter	Reference number: Invalid Parameter	“Ensure that Order Status OR Order Type parameters are not invalid.”	UpdateBillingDetails
51319	Invalid Parameter	bill_id: Invalid Parameter	“Ensure that bill_id parameter is not invalid.”	UpdateInvoiceReferenceNumber
51320	Invalid Parameter	Bill Merchant Reference No: Invalid Parameter	“Bill Merchant Reference No parameter already exists.”	UpdateInvoiceReferenceNumber
51321	Invalid Parameter	Reg ID: Invalid Parameter	“Ensure that Reg ID parameter is not invalid.”	UpdateMerchantParams
51322	Invalid Parameter	Reference number: Invalid Parameter	"Ensure that reference number parameter is not invalid."	UpdateMerchantParams
51323	Invalid Parameter	Tax List: Invalid Parameter	"Ensure that Tax is configured for the merchant."	GenerateInvoice ,GenerateRecurringInvoice
51324	Invalid Parameter	Tax List: Invalid Parameter	“Ensure that Tax value matches with the given Tax name.”	GenerateInvoice ,GenerateRecurringInvoice
51325	Missing Parameter	Customer id: Required parameter missing	“Ensure that Customer parameter is present.”	DeleteCustomer,
51326	Invalid Parameter	Currency: Invalid Parameter	"Ensure that Currency is assigned to the merchant."	GenerateInvoice, GenerateRecurringInvoice, GenerateQuickInvoice
51327	Invalid Parameter	Customer id: Invalid Parameter	"Ensure that the merchant is not invalid."	OrderLookup, Delete Customer
51328	Invalid Parameter	Reference number: Invalid Parameter	"Multiple refunds not allowed."	RefundOrder
51336	Invalid Parameter	Customer Card Id: Invalid Parameter	"Card Not Present."	DeleteCustomer payment option