CCAvenue Integration Document

Version 3.4

INDEX

Contents

CCAvenue Integration Document

Introduction

CCAvenue payment integration kit allows merchants to instantly collect payments from their users using various payment modes like credit cards, debit cards, cash cards, net banking etc.

The CCAvenue payment integration supports a seamless payment experience on your platform, while protecting your application from payment frauds and complexity related to various regulations.

Testing and Production Environment

CCAvenue test and production environments are separate.

Merchants need an active CCAvenue account to use the test environment and production environment. Merchants will have to log in to their CCAvenue M.A.R.S account and get the API credentials for using these environments.

All transactions initiated by the merchant on our test environment are not processed. Test environment is strictly for testing the request and response functions.

After successfully testing the integration, merchant can move to the production environment by changing the URL.

CCAvenue TEST URL is:
https://test.ccavenue.com/transaction/transaction.do?command=initiateTransaction

CCAvenue PRODUCTION URL is:
https://secure.ccavenue.com/transaction/transaction.do?command=initiateTransaction

To test the integration login to your CCAvenue M.A.R.S account, under Settings tab -> API Keys page; copy the following credentials:

1. Merchant ID
2. Access Code
3. Working Key

Integration Methods

CCAvenue supports collecting payment information using following methods. All methods are designed to support a seamless user-experience.

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.

2. **Custom checkout form (Seamless) -** Merchants can build a custom checkout form to collect order and payment information and pass 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. To enable Seamless feature, we required PCIDSS certificate from merchant.

   TokenPay: This integration enables merchants to securely process the end-customer's card without actually storing the entire details and only storing the surrogate value of the same. It works across all major card networks, including MasterCard, RuPay, and Visa.

3. **CCAvenue "Direct Connect" (Payload) -** This integration enables you to deliver payment services directly through your website without redirecting your users to CCAvenue. This integration is fast and secure. It gives you control to not only build your own custom checkout form, but also control the payment request process with the banks.

   TokenPay: This integration enables merchants to securely process the end-customer's card without actually storing the entire details and only storing the surrogate value of the same. It works across all major card networks, including MasterCard, RuPay, and Visa.

Processing orders using CCAvenue billing page (Non-Seamless)

Processing orders using CCAvenue billing page

CCAvenue billing page helps you avoid the hassle of developing and managing your own billing page. CCAvenue billing page is fully customizable enabling you to match the look and feel of your website.

Process flow

1. Customer selects product/service on your website and proceeds to make payment.

2. Customer is redirected to the CCAvenue billing page where billing, shipping and payment information is entered by the customer.

3. On submission of the transaction information, CCAvenue initiates the authorization process by connecting to the relevant bank/processing organization.

4. On receiving the authorization status from the bank, CCAvenue sends the response back to your website with the transaction status.

Basic steps involved in integration with the CCAvenue billing page:

Set Up: Download the CCAvenue client library from the MARS panel. Click on “Resources” on the navigation bar of the Dashboard and click “Integration Kit”. You will have to use the CCAvenue transaction file (e.g. ccavRequestHandler.php) to initiate the payment process.

Configure: Every merchant receives a unique set of keys for transaction processing. These need to be configured in the transaction file used to initiate the payment process**.**

From your MARS account under Settings tab -> API Keys page; copy the merchant id, access code and secret encryption. Set these values in the file (e.g. ccavRequestHandler.php) downloaded with the integration kit.

Payment Processing: You will have to post the order information to the CCAvenue transaction file (e.g. ccavRequestHandler.jsp) to initiate the payment process. CCAvenue transaction file on receiving the order related data will encrypt the data and forward the encrypted request to the CCAvenue billing page.

JSP

```
<html>
<head><title>Sample Transaction File</title></head>
<body>
<%@ page import = "java.io.*, com.ccavenue.transaction.util.AesCryptUtil" %>
<%@include file="libFunctions.jsp"%>
<%
  String merchant_id = "2193";  //Put your merchant id here
  String access_code = " F94007DF1640D69A";     //Put access code here
  String enc_key = "FABE114254BDBC7823534894FFFCCC1";     //Put encryption key here
  Enumeration enumeration=request.getParameterNames ();
  String ccaRequest="", pname="", pvalue="";
   while (enumeration.hasMoreElements ()) {
    pname = ""+enumeration.nextElement ();
     pvalue = request.getParameter (pname);
     ccaRequest = ccaRequest + pname + "=" + pvalue + "&";
  }
    AesCryptUtil aesUtil=new AesCryptUtil (enc_key);
  String encRequest=aesUtil.encrypt (ccaRequest);
%>
<form method="post" name="redirect"
action="https://test.ccavenue.com/transaction/transaction.do? command= initiateTransaction"/>
<input type="hidden" id="encRequest" name="encRequest" value="<%= encRequest %>">
<input type="hidden" name="access_code" id="access_code" value="<%= access_code %>">
<script language='javascript'>document.redirect.submit ();</script>
</form>
</body>
 </html>
```

Request Parameters

Merchant must send the following parameters to the CCAvenue PG for processing an order.

Required Parameters		
Name	**Description**	**Type (length)**
merchant_id	Merchant Id is a unique identifier generated by CCAvenue for each activated merchant.	Numeric
order_id	This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. **CCAvenue will not check the uniqueness of this order id.** As it generates a unique payment reference number for each order which is sent by the merchant.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
currency	Currency in which you want to process the transaction. INR – Indian Rupee USD – United States Dollar SGD – Singapore Dollar GBP – Pound Sterling EUR – Euro, official currency of Eurozone	Alphabets (3)
amount	Order amount	Numeric (12, 2)
redirect_url	CCAvenue will post the status of the order along with the parameters to this URL. If you do not send this value, order status will be sent back to the URL configured in dynamic event notifications module in your MARS account. If there is no URL configured in the MARS account, PG will display the status of the order on the CCAvenue confirmation page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
cancel_url	CCAvenue will redirect the customer to this URL if the customer cancels the transaction on the billing page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
language	CCAvenue billing page is multi-lingual. Currently we are displaying the page in English (Code - EN).	Alphabet(5)

Merchant can send any of the following parameters in addition to the required parameters.

Billing and Shipping Information		
Name	**Description**	**Type (length)**
billing_name	Name of the customer	Alphabets (60) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_address	Customer's billing address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
billing_city	Customer's billing city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_state	Customer's billing state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_zip	Customer's billing zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
billing_country	Customer's billing country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_tel	Customer's phone number	Numeric (20)

billing_email	Customer's email address	Alphanumeric (70) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers @ (at), dot,_ (underscore)
delivery_name	Recipient's name	Alphabets (60) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_address	Shipping address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
delivery_city	Shipping city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_state	Shipping state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_zip	Shipping zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
delivery_country	Shipping country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.

delivery_tel	Shipping phone number	Numeric (20)
merchant_param1	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param2	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param3	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param4	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param5	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (5000) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
promo_code	This parameter is used for sending the code of the promotion you have created in the CCAvenue MARs by which you may offer specific discounts to customers using specific payment options.	Alphanumeric (20) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers

tid	This parameter is used for sending the unique identifier to identify uniqueness of the order. This is an optional parameter. Value for this parameter can be generated using the piece of code given in the integration kit. ***The uniqueness of TID is valid for 24 hours only.***	Numeric(17) **Characters allowed:** Only numbers
sub_account_id **(Mandatory for sub_account_id integration only)**	This parameter can be used for sending sub account id which is configured against your merchant id. Only single sub account is allowed at a time to pass and settlement will get processed in the same sub account id.	Alphanumeric (which is configured against the merchant id)

Request parameters for Standing Instruction Information

Merchant must send the following parameters to the CCAvenue PG for setting Standing Instructions for customer.

si_type *(required)*	This parameter is used to identify whether the standing instruction request is for the fixed amount or for variable amount. Expected values: Fixed Variable	Alphabet(8)
si_mer_ref_no	This parameter can be used by the merchant to send a unique identifier. E.G. For insurance – Policy number. It can also be a customer reference number.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
si_amount	This will be required only in case of “Fixed” type standing instruction. This SI amount will be charged to the customer on each billing cycle.	Decimal (12,2)
si_setup_amount	This is a mandatory field and is required as part of the SI creation process. This is a one-time charge	
si_frequency	This will be required only in case of “Fixed” type standing instruction. Expected values: Week Month Year This is used with si_frequency_no. E.g. If you want to charge the customer every 2 months, you will set the si_frequency parameter as “Month” and si_frequency_no as 2.	Alphabet(5)

si_frequency_no	This will be required only in case of “Fixed” type standing instruction. This parameter will enable you to set the frequency on which you want to charge the customer. This is used with si_frequency. E.g. If you want to charge the customer every 2 months, you will set the si_frequency parameter as “Month” and si_frequency_no as 2.	Numeric
si_ billing_cycle	This will be required only in case of “Fixed” type standing instruction. This parameter will enable you to set the value for total number of times you want to charge a customer. E.g. If you want to charge the customer 10 times every 2 months, you will set the si_frequency as “Month”, si_frequency_no as 2 and si_billing_cycle as 10.	Numeric
si_start_date	This will be required only in case of “Fixed” type standing instructions. This is the date from which SI billing will start for the customer.	datetime

***Disclaimer: “billing_zip” parameter is “Mandatory” for Insurance domain merchants creating SI profile with AMEX card scheme.**

Response parameters for Tokenization

Name	Description	Type (length)
token_eligibility	Token Eligibility	Alphabet(1) Values provided will be (Y/N)

Request Handling:

Please find the below sample of Decrypted encRequest(Query string) which merchant needs to generate after clicking on the final checkout button to get redirected on payment page.

Decrypted encRequest:

merchant_id=123456&order_id=1234578&currency=INR&amount=1.00&redirect_url=https://www.abc.com/&cancel_url=https://www.abc.com

The above decrypted encRequest needs to be encrypted with AES 128 bit algorithm which is provided in the integration kit. Merchant needs to encrypt the parameters using correct working key.
Please find the below final request url which merchant needs to generate to redirect on payment page.

Sample request (UAT/Testing):
https://test.ccavenue.com/transaction/transaction.do?command=initiateTransaction&encRequest=ccee6b03e00fa4f3657cf56a57bc681341cede3be1c5a077510f239511efa413dfd5ac3dacdbe0ddd2ae6806fc9c04229be8b4535896b0cab939ccd4ec9a9560&access_code=AVGP00ZZ24BB27AAAA

Sample request (PROD):
https://secure.ccavenue.com/transaction/transaction.do?command=initiateTransaction&encRequest=ccee6b03e00fa4f3657cf56a57bc681341cede3be1c5a077510f239511efa413dfd5ac3dacdbe0ddd2ae6806fc9c04229be8b4535896b0cab939ccd4ec9a9560&access_code=AVGPUUZZ24BB27BBBB

Processing orders using custom checkout form

Merchants can build a custom checkout form to collect order and payment information and pass the same to CCAvenue directly for payment processing.

Process flow

1. Customer after selecting the product/service and entering the shipping details will proceed to make the payment on your billing page.

2. On your customized billing page customer selects the payment option from the list provided by CCAvenue as a JSON object. Customer enters the payment information and submits the form.

3. On submission of the payment information, CCAvenue initiates the authorization process by connecting to the relevant bank/processing organization.

4. On receiving the authorization status from the bank, CCAvenue sends the response back to your website with the transaction status.

Basic steps involved in fetching payment options to create your custom checkout form:

Set Up: Download the CCAvenue client library from the MARS panel. Click on “Resources” on the navigation bar of the Dashboard and click “Integration Kit”. You will have to use the CCAvenue transaction file (e.g. ccavRequestHandler.php) to initiate the payment process.

Configure: Every merchant receives a unique set of keys for transaction processing. These need to be configured in the transaction file used to initiate the payment process**.**

From your MARS account under Settings tab -> API Keys page; copy the merchant id, access code and secret encryption. Set these values in the file (e.g. ccavRequestHandler.php) downloaded with the integration kit.

Payment Processing: You will have to post the order information to the CCAvenue transaction file (e.g. ccavRequestHandler.jsp) to initiate the payment process. CCAvenue transaction file on receiving the order related data will encrypt the data and forward the encrypted request to the CCAvenue billing page.

JSON object will contain following information:

1. **Payment Option Type –** Will contain payment options allocated to the merchant. Options may include Credit Card, Net Banking, Debit Card, Cash Cards, EMI Payments or Mobile Payments.

2. **Card Type –** Will contain card type allocated to the merchant. Options may include Credit Card, Net Banking, Debit Card, Cash Cards or Mobile Payments.

3. **Card Name –** Will contain name of card. E.g. Visa, MasterCard, American Express or and bank name in case of Net banking.

4. **Payment Mode Status –** Will help in identifying the status of the payment mode. Options may include Active or Down.

5. **Error –** This parameter will enable you to troubleshoot any configuration related issues. It will provide error description.

You will have to post the order information to the CCAvenue transaction file (e.g. ccavRequestHandler.jsp) to initiate the payment process. CCAvenue transaction file on receiving the order related data will encrypt the data and forward the encrypted request to the CCAvenue server for processing.

Sample Code

```
<script type="text/javascript">
$(function(){
    var jsonData;
    var access_code=""; //Put access code here
    var amount="10.00";
    var currency="INR";

    $.ajax({
        url:'https://test.ccavenue.com/transaction/transaction.do?command=getJsonData&access_code='+
             access_code+'&currency='+currency+'&amount='+amount,
        dataType: 'jsonp',
        jsonp: false,
        jsonpCallback: 'processData',
        success: function (data) {
            jsonData = data;
        },
        error: function(xhr, textStatus, errorThrown) {
            alert('An error occurred! ' + ( errorThrown ? errorThrown :xhr.status ));
        }
    });

    $(".payOption").click(function(){
        $("#card_name").children().remove(); // remove old card names from old one
        $("#card_name").append("<option value=''>Select</option>");

        var paymentOption = $(this).val();
        $("#card_type").val(paymentOption.replace("OPT",""));

        $.each(jsonData, function(index,value) {
            if(value.payOpt==paymentOption){
                var payOptJSONArray = $.parseJSON(value[paymentOption]);
                $.each(payOptJSONArray, function() {
                    $("#card_name").find("option:last").after("<option class='"+this['dataAcceptedAt']+" "+
                    this['status']+"'  value='"+this['cardName']+"'>"+this['cardName']+"</option>");
                });
            }
        });
    });

    $("#card_name").click(function(){
        if($(this).find(":selected").hasClass("DOWN")){
            alert("Selected option is currently unavailable. Select another payment option or try again later.");
        }
        if($(this).find(":selected").hasClass("CCAvenue")){
            $("#data_accept").val("Y");
        }else{
            $("#data_accept").val("N");
        }
    });
</script>
```

Request Parameters

Merchant must send the following parameters to the CCAvenue PG for processing an order.

Required Parameters		
Name	**Description**	**Type (length)**
merchant_id	Merchant Id is a unique identifier generated by CCAvenue for each activated merchant.	Numeric
order_id	This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. **CCAvenue will not check the uniqueness of this order id.** As it generates a unique payment reference number for each order which is sent by the merchant.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
currency	Currency in which you want to process the transaction. INR – Indian Rupee USD – United States Dollar SGD – Singapore Dollar GBP – Pound Sterling EUR – Euro, official currency of Eurozone	Alphabets (3)
amount	Order amount	Numeric (12, 2)
redirect_url	CCAvenue will post the status of the order along with the parameters to this URL. If you do not send this value, order status will be sent back to the URL configured in dynamic event notifications module in your MARS account. If there is no URL configured in the MARS account, PG will display the status of the order on the CCAvenue confirmation page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
cancel_url	CCAvenue will redirect the customer to this URL if the customer cancels the transaction on the billing page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)

payment_option	Payment option selected by the customer OPTCRDC - Credit Card OPTDBCRD - Debit Card OPTNBK - Net Banking OPTCASHC - Cash Card OPTMOBP - Mobile Payments	Alphabets (10)
card_type	Type of card used by the customer. CRDC - Credit Card DBCRD - Debit Card NBK - Net Banking CASHC - Cash Card MOBP - Mobile Payments	Alphabets (10)
card_name	Name of the card used by the customer. This list will be provided by CCAvenue.	Alphabets(100) **Characters allowed:** Alphabet (A-Z), (a-z).
data_accept	Resend the parameter value received at the time of fetching the payment options. **Expected values –** Y or N	Alphabets(1)
card_number	Card number entered by the customer. ** (Please refer **Note** at the end of this table)*	Numeric
expiry_month	Card expiry month ** (Please refer **Note** at the end of this table)*	Numeric
expiry_year	Card expiry year ** (Please refer **Note** at the end of this table)*	Numeric
cvv_number	Card CVV number	Numeric
issuing_bank	Card issuing bank name	Alphabets(100) **Characters allowed:** Alphabet (A-Z), (a-z).
mobile_no	Mobile no (Only in case of Mobile payments.)	Numeric

*** Note: This parameter is required ONLY in case of card-in-motion. If the card is tokenised by the merchant, please refer "Request parameters for Tokenised Transaction" section.**

Merchant can send any of the following parameters in addition to the required parameters.

Name	Description	Type (length)
billing_name	Name of the customer	Alphabets (60) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_address	Customer's billing address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
billing_city	Customer's billing city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_state	Customer's billing state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_zip	Customer's billing zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
billing_country	Customer's billing country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_tel	Customer's phone number	Numeric (20)

billing_email	Customer's email address	Alphanumeric (70) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers @ (at), dot,_ (underscore)
delivery_name	Recipient's name	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_address	Shipping address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
delivery_city	Shipping city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_state	Shipping state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_zip	Shipping zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
delivery_country	Shipping country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_tel	Shipping phone number	Numeric (20)

merchant_param1	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param2	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param3	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param4	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param5	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (5000) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)

sub_account_id **(Mandatory for sub_account_id integration only)**	This parameter can be used for sending sub account id which is configured against your merchant id. Only single sub account is allowed at a time to pass and settlement will get processed in the same sub account id.	Alphanumeric (which is configured against the merchant id)

Request parameters for Standing Instruction Information

Merchant must send the following parameters to the CCAvenue PG for setting Standing Instructions for customer.

si_type *(required)*	This parameter is used to identify whether the standing instruction request is for the fixed amount or for variable amount. Expected values: Fixed Variable	Alphabet(8)
si_mer_ref_no	This parameter can be used by the merchant to send a unique identifier. E.G. For insurance – Policy number. It can also be a customer reference number.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
si_amount	This will be required only in case of “Fixed” type standing instruction. This SI amount will be charged to the customer on each billing cycle.	Decimal (12,2)
si_setup_amount	This is a mandatory field and is required as part of the SI creation process. This is a one-time charge	Decimal (12,2)
si_frequency	This will be required only in case of “Fixed” type standing instruction. Expected values: Week Month Year This is used with si_frequency_no. E.g. If you want to charge the customer every 2 months, you will set the si_frequency parameter as “Month” and si_frequency_no as 2.	Alphabet(5)

si_frequency_no	This will be required only in case of “Fixed” type standing instruction. This parameter will enable you to set the frequency on which you want to charge the customer. This is used with si_frequency. E.g. If you want to charge the customer every 2 months, you will set the si_frequency parameter as “Month” and si_frequency_no as 2.	Numeric
si_ billing_cycle	This will be required only in case of “Fixed” type standing instruction. This parameter will enable you to set the value for total number of times you want to charge a customer. E.g. If you want to charge the customer 10 times every 2 months, you will set the si_frequency as “Month”, si_frequency_no as 2 and si_billing_cycle as 10.	Numeric
si_start_date	This will be required only in case of “Fixed” type standing instructions. This is the date from which SI billing will start for the customer.	datetime

***Disclaimer: “billing_zip” parameter is “Mandatory” for Insurance domain merchants creating SI profile with AMEX card scheme.**

Request parameters for Tokenised Transaction

Merchant must pass the below set of parameters to process a tokenised card in case if the token was NOT provisioned through CCAvenue TokenPay.

Required Parameters		
Name	**Description**	**Type (length)**
cvv	**Card verification value number**	**Numeric(5)**
cryptogram	Cryptogram values	Alphanumeric(40)
token_requestor_id	Token requestor ID	Alphanumeric(40)
token_number	Token number	Alphanumeric(40)
token_expiry	Token expiry date	Numeric(10) Values allowed (mm/yyyy)

Disclaimer: Parameter "cryptogram" will hold "DCSC" code which is 4 digit numeric value for the AMEX card scheme.

NOTE: In case the merchant has opted for CCAvenue TokenPay - Provisioning Services, our application will do the heavy lifting and will fetch all above information for payment processing.

Response Parameters for Tokenization

Name	Description	Type (length)
token_eligibility	Token Eligibility	Alphabet(1) Values provided will be (Y/N)

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			
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

Processing orders using CCAvenue Direct Connect (Payload)

This integration will enable you to deliver payment services directly through your website without redirecting your users to CCAvenue. This integration is fast and secure. It gives you control to not only build your own custom checkout form, but also control the payment request process with the banks.

Process flow

1. Customer after selecting the product/service and entering the shipping details will proceed to make the payment using your billing page.

2. On your customized billing page customer selects the payment option from the list provided by CCAvenue as a JSON object. Customer enters the payment information and submits the form.

3. 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.

4. The request payload received from CCAvenue will be used by you to connect directly to the bank's authentication/3D secure page, bypassing CCAvenue.

5. CCAvenue will receive the authentication status from the bank and in turn post the transaction status back to the merchant's website.

Basic steps involved in fetching payment options to create your custom checkout form:

Set Up: Download the CCAvenue client library from the MARS panel. Click on “Resources” on the navigation bar of the Dashboard and click “Integration Kit”. You will have to use the CCAvenue transaction file (e.g. ccavRequestHandler.php) to initiate the payment process.

Configure: Every merchant receives a unique set of keys for transaction processing. These need to be configured in the transaction file used to initiate the payment process**.**

From your MARS account under Settings tab -> API Keys page; copy the merchant id, access code and secret encryption. Set these values in the file (e.g. ccavRequestHandler.php) downloaded with the integration kit.

Payment Processing: You will have to post the order information to the CCAvenue transaction file (e.g. ccavRequestHandler.jsp) to initiate the payment process. CCAvenue transaction file on receiving the order related data will encrypt the data and forward the encrypted request to the CCAvenue billing page.

JSON object will contain following information:

1. **Payment Option Type –** Will contain payment options allocated to the merchant. Options may include Credit Card, Net Banking, Debit Card, Cash Cards, EMI Payments or Mobile Payments.

2. **Card Type –** Will contain card type allocated to the merchant. Options may include Credit Card, Net Banking, Debit Card, Cash Cards or Mobile Payments.

3. **Card Name –** Will contain name of card. E.g. Visa, MasterCard, American Express or and bank name in case of Net banking.

4. **Payment Mode Status –** Will help in identifying the status of the payment mode. Options may include Active or Down.

5. **Error –** This parameter will enable you to troubleshoot any configuration related issues. It will provide error description.

You will have to post the order information to the CCAvenue transaction file (e.g. ccavRequestHandler.jsp) to initiate the payment process. CCAvenue transaction file on receiving the order related data will encrypt the data and forward the encrypted request to the CCAvenue server for processing.

Sample Code

```
<script type="text/javascript">
$(function(){
    var jsonData;
    var access_code=""; //Put access code here
    var amount="10.00";
    var currency="INR";

    $.ajax({
        url:'https://test.ccavenue.com/transaction/transaction.do?command=getJsonData&access_code='+
             access_code+'&currency='+currency+'&amount='+amount,
        dataType: 'jsonp',
        jsonp: false,
        jsonpCallback: 'processData',
        success: function (data) {
            jsonData = data;
        },
        error: function(xhr, textStatus, errorThrown) {
            alert('An error occurred! ' + ( errorThrown ? errorThrown :xhr.status ));
        }
    });

    $(".payOption").click(function(){
        $("#card_name").children().remove(); // remove old card names from old one
        $("#card_name").append("<option value=''>Select</option>");

        var paymentOption = $(this).val();
        $("#card_type").val(paymentOption.replace("OPT",""));

        $.each(jsonData, function(index,value) {
            if(value.payOpt==paymentOption){
                var payOptJSONArray = $.parseJSON(value[paymentOption]);
                $.each(payOptJSONArray, function() {
                    $("#card_name").find("option:last").after("<option class='"+this['dataAcceptedAt']+" "+
                    this['status']+"'  value='"+this['cardName']+"'>"+this['cardName']+"</option>");
                });
            }
        });
    });

    $("#card_name").click(function(){
        if($(this).find(":selected").hasClass("DOWN")){
            alert("Selected option is currently unavailable. Select another payment option or try again later.");
        }
        if($(this).find(":selected").hasClass("CCAvenue")){
            $("#data_accept").val("Y");
        }else{
            $("#data_accept").val("N");
        }
    });
</script>
```

Sample code for handling Payload at your end:

For HTML

```
HttpClient vClient = new HttpClient();
String vResponse =
vClient.processUrlConnectionReq("encRequest="+encRequest+"&access_code="+access_code,"https:/
/test.ccavenue.com/transaction/transaction.do?command=initiatePayloadTransaction");
out.print(vResponse);//writes payload on browser to open bank page
```

For JSON:

```
HttpClient vClient = new HttpClient();
String vResponse =
vClient.processUrlConnectionReq("encRequest="+encRequest+"&access_code="+access_code,"https:/
/test.ccavenue.com/transaction/transaction.do?command=initiatePayloadTransaction");//makes
server to server call to CCAvenue and recives payload in JSON fromat
String vHtml = "<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'
'http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd'>"
                                        +"<html xmlns='http://www.w3.org/1999/xhtml'>"
                                        + "<head>" + "<meta http-equiv='Content-Type' content='text/html;
charset=utf-8' />" + "<title>CCAvenue-Transaction page</title>" + "<link rel='SHORTCUT ICON'
type='image/ico' href='"
                                + "/images/favicon.ico' />"
                                        +"<script language='javascript'>window.history.forward(); function
noBack() { window.history.forward(); } function SubmitMe(){
document.getElementById('submit').style.visibility='hidden';document.getElementById('submit').click()
; }</script>"
                                        + "</head>"
                                        + "<body style='margin:0px;' onLoad='noBack();SubmitMe();'>";
  JSONObject obj = new JSONObject(vResponse);
  vHtml=vHtml+"<form name='MalltoEpay' method='"+obj.get("method")+"'
action='"+obj.get("bankUrl")+"'>";
  JSONObject requestData = obj.getJSONObject("data");
  Iterator vKeys = requestData.keys();
  while(vKeys.hasNext()){
  String key = (String)vKeys.next();
          vHtml = vHtml+"<input type='text' name='"+key+"' value='"+requestData.get(key)+"'>"; }
  vHtml = vHtml+"<input type='submit' id='submit' value='Continue' style='display:none;'></form>"
  +"</body></html>";
  out.print(vHtml);
```

Request Parameters

Merchant must send the following parameters to the CCAvenue PG for processing an order.

Required Parameters		
Name	**Description**	**Type (length)**
merchant_id	Merchant Id is a unique identifier generated by CCAvenue for each activated merchant.	Numeric
order_id	This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. **CCAvenue will not check the uniqueness of this order id.** As it generates a unique payment reference number for each order which is sent by the merchant.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
currency	Currency in which you want to process the transaction. INR – Indian Rupee USD – United States Dollar SGD – Singapore Dollar GBP – Pound Sterling EUR – Euro, official currency of Eurozone	Alphabets (3)
amount	Order amount	Numeric (12, 2)
redirect_url	CCAvenue will post the status of the order along with the parameters to this URL. If you do not send this value, order status will be sent back to the URL configured in dynamic event notifications module in your MARS account. If there is no URL configured in the MARS account, PG will display the status of the order on the CCAvenue confirmation page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
cancel_url	CCAvenue will redirect the customer to this URL if the customer cancels the transaction on the billing page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)

payment_option	Payment option selected by the customer OPTCRDC - Credit Card OPTDBCRD - Debit Card OPTNBK - Net Banking OPTCASHC - Cash Card OPTMOBP - Mobile Payments	Alphabets (10)
card_type	Type of card used by the customer. CRDC - Credit Card DBCRD - Debit Card NBK - Net Banking CASHC - Cash Card MOBP - Mobile Payments	Alphabets (10)
card_name	Name of the card used by the customer. This list will be provided by CCAvenue.	Alphabets(100) **Characters allowed:** Alphabet (A-Z), (a-z).
data_accept	Resend the parameter value received at the time of fetching the payment options. **Expected values –** Y or N	Alphabets(1)
card_number	Card number entered by the customer. ** (Please refer **Note** at the end of this table)*	Numeric
expiry_month	Card expiry month ** (Please refer **Note** at the end of this table)*	Numeric
expiry_year	Card expiry year ** (Please refer **Note** at the end of this table)*	Numeric
cvv_number	Card CVV number	Numeric
issuing_bank	Card issuing bank name	Alphabets(100) **Characters allowed:** Alphabet (A-Z), (a-z).
mobile_no	Mobile no (Only in case of Mobile payments.)	Numeric

*** Note: This parameter is required ONLY in case of card-in-motion. If the card is tokenised by the merchant, please refer "Request parameters for Tokenised Transaction" section.**

Merchant can send any of the following parameters in addition to the required parameters.

Name	Description	Type (length)
billing_name	Name of the customer	Alphabets (60) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_address	Customer's billing address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
billing_city	Customer's billing city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_state	Customer's billing state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_zip	Customer's billing zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
billing_country	Customer's billing country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_tel	Customer's phone number	Numeric (20)

billing_email	Customer's email address	Alphanumeric (70) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers @ (at), dot,_ (underscore)
delivery_name	Recipient's name	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_address	Shipping address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
delivery_city	Shipping city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_state	Shipping state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_zip	Shipping zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
delivery_country	Shipping country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_tel	Shipping phone number	Numeric (20)

device_parameter	This optional parameter is used only in case Direct Connect integration for in which merchant sends a device type though which transaction is processed.	Alphabets (3) **Characters allowed:** MO BPC
merchant_param1	This parameter can be used for sending additionalinformation about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param2	This parameter can be used for sending additionalinformation about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param3	This parameter can be used for sending additionalinformation about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param4	This parameter can be used for sending additionalinformation about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param5	This parameter can be used for sending additionalinformation about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (5000) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)

Request parameters for Standing Instruction Information

Merchant must send the following parameters to the CCAvenue PG for setting Standing Instructions for customer.

si_type *(required)*	This parameter is used to identify whether the standing instruction request is for the fixed amount or for variable amount. Expected values: Fixed Variable	Alphabet(8)
si_mer_ref_no	This parameter can be used by the merchant to send a unique identifier. E.G. For insurance – Policy number. It can also be a customer reference number.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
si_amount	This will be required only in case of “Fixed” type standing instruction. This SI amount will be charged to the customer on each billing cycle.	Decimal (12,2)
si_setup_amount	This is a mandatory field and is required as part of the SI creation process. This is a one-time charge	
si_frequency	This will be required only in case of “Fixed” type standing instruction. Expected values: Week Month Year This is used with si_frequency_no. E.g. If you want to charge the customer every 2 months, you will set the si_frequency parameter as “Month” and si_frequency_no as 2.	Alphabet(5)

si_frequency_no	This will be required only in case of “Fixed” type standing instruction. This parameter will enable you to set the frequency on which you want to charge the customer. This is used with si_frequency. E.g. If you want to charge the customer every 2 months, you will set the si_frequency parameter as “Month” and si_frequency_no as 2.	Numeric
si_ billing_cycle	This will be required only in case of “Fixed” type standing instruction. This parameter will enable you to set the value for total number of times you want to charge a customer. E.g. If you want to charge the customer 10 times every 2 months, you will set the si_frequency as “Month”, si_frequency_no as 2 and si_billing_cycle as 10.	Numeric
si_start_date	This will be required only in case of “Fixed” type standing instructions. This is the date from which SI billing will start for the customer.	datetime

***Disclaimer: “billing_zip” parameter is “Mandatory” for Insurance domain merchants creating SI profile with AMEX card scheme.**

Request parameters for Tokenised Transaction

Merchant must pass the below set of parameters to process a tokenised card in case if the token was NOT provisioned through CCAvenue TokenPay.

Required Parameters		
Name	**Description**	**Type (length)**
cvv	**Card verification value number**	**Numeric(5)**
cryptogram	Cryptogram values Required only if tokenization not done through CCAvenue	Alphanumeric(40)
token_requestor_id	Token requestor ID	Alphanumeric(40)
token_number	Token number	Alphanumeric(40)
token_expiry	Token expiry date	Numeric(10)

		Values allowed (mm/yyyy)

Disclaimer: Parameter "cryptogram" will hold "DCSC" code which is 4 digit numeric value for the AMEX card scheme.

NOTE: In case the merchant has opted for CCAvenue TokenPay - Provisioning Services, our application will do the heavy lifting and will fetch all above information for payment processing.

Response Parameters for Tokenization

Name	Description	Type (length)
token_eligibility	Token Eligibility	Alphabet(1) Values provided will be (Y/N)

Response Parameters

CCAvenue PG will return following parameters:

Name	Description	Type (length)
order_id	This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. **CCAvenue will not check the uniqueness of this order id.** As it generates a unique payment reference number for each order which is sent by the merchant Kindly ensure this value received in response is validated before providing services.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, # (hash), /(slash, - (hyphen)
tracking_id	Unique payment reference number generated by CCAvenue for each order.	Numeric (12)
bank_ref_no	Reference number generated by the bank for the transaction.	Alphanumeric
order_status	Status of the order. Success Failure Aborted Invalid Timeout	Alphabets (15)
failure_message	Reason for failure.	Alphanumeric
payment_mode	The payment mode used by customer IVRS EMI Credit Card Net banking Debit Card Cash Card UPI Wallet	Alphabets
card_name	Specifies the type of credit card, debit card, netbanking etc .	Alphanumeric
status_code	The status code for this transaction	Numeric (3)
status_message	The status message for this transaction.	Alphanumeric (150)

currency	Currency code in which the transaction was processed. INR – Indian Rupee USD – United States Dollar SGD – Singapore Dollar GBP – Pound Sterling EUR – Euro, official currency of Eurozone Kindly ensure this value received in response is validated before providing services.	Alphabets (3)
Amount	Order amount Kindly ensure this value received in response is validated before providing services.	Numeric (12, 2)
billing_name	Name of the customer	Alphabets (60) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_address	Customer's billing address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
billing_city	Customer's billing city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_state	Customer's billing state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.

billing_zip	Customer's billing zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
billing_country	Customer's billing country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_tel	Customer's phone number	Numeric (20)
billing_email	Customer's email address	Alphanumeric (70) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers @ (at), dot,_ (underscore)
delivery_name	Recipient's name	Alphabets (60) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_address	Shipping address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.

delivery_city	Shipping city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_state	Shipping state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_zip	Shipping zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
delivery_country	Shipping country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_tel	Shipping phone number	Numeric (22)
merchant_param1	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param2	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)

merchant_param3	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param4	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (500) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param5	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (5000) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
vault	This parameter can be used if merchant availing the vault option. On using vault functionality if card details are saved at CCAvenue end value returned will be Y. If card details are not saved at CCAvenue end the value returned for this parameter will be N	Character(1) **Characters allowed:** Y or N
offer_type	This parameter can be used for sending additional information if customer has used any discount or promotion while completing the transaction. If customer is using discount-coupon, value of this parameter would be discount. If customer is using promo-code, value of this parameter would be promotion.	Alphabets (9)

offer_code	This parameter can be used for sending additional information about the discount coupon and Promo code used while completing the transaction. If customer has used Discount the value sent would be Y or N accordingly. If customer has used Promotion the value sent would be Promo code	Alphanumeric (30)
discount_value	This parameter can be used for sending additional information about the discounted amount.	Numeric (12,2)
si_status	Status of the standing instruction request "0" denotes success. "1" denotes failure. This parameter is applicable for only for SI transactions.	Numeric
si_sub_ref_no	This is reference number created by CCAvenue for each new subscription on the CCAvenue system. This is the number that must be sent with each new "on demand" charge to identify the customer.	Alphanumeric (15)
si_mer_ref_no	This is the unique identifier send by the merchant in the request. E.G. For insurance – Policy number. It can also be a customer reference number.	Alphanumeric (30)
si_error_desc	Reason for failure to setup SI.	Alphanumeric (150)
si_created	SI is created or not (Optional in case of SI only) Value: Y - SI created N - SI not created	Character (1)
si_ref_no	SI Reference Number (Optional in case of SI only)	Alphanumeric (15)
retry	This parameter can be used if merchant availing the retry option. If the transaction is processed through retry attempt returned value will be Y. If the transaction is not processed through retry attempt returned value will be n.	Character(1) **Characters allowed:** Y or N

response_code	This parameter contains the code for each bank response message.	Numeric
bene_account	NEFT client code + tracking id (Optional in case of NEFT only)	Alphanumeric (35)
bene_name	NEFT client code (Optional in case of NEFT only)	Alphanumeric (20)
bene_ifsc	Beneficiary IFSC code (Optional in case of NEFT only)	Alphanumeric (20)
bene_bank	Beneficiary Bank code (Optional in case of NEFT only)	Alphanumeric (50)
bene_branch	Beneficiary Bank Branch (Optional in case of NEFT only)	Alphanumeric (255)
inv_mer_reference_no	Merchant reference number of invoice (Optional in case of invoice transaction only)	Alphanumeric (100)
trans_date	Transaction Completion Date	DateTime **dd/MM/yyyy HH:mm:ss**
mer_amount	In case of charge to customer model this amount is paid to the merchant.	Numeric (12, 2)
sub_account_id	This parameter returns the Sub Account ID sent by merchant while initiating the transaction.	Alphanumeric (20)
eci_value	ECI value as received from 3 D secure.	Numeric (2)
billing_notes	This parameter returns the billing notes entered by customer on the billing page.	Alphanumeric (150) Only letters, numbers, dot, &, circular brackets, slash, comma and hyphen are allowed.
bin_country	This parameter returns the entered Credit or Debit cards BIN country.	Alphanumeric(255)
customer_card_id	The identifier against which the card information is stored or retrieved. This is used in case of Vault transactions.	Numeric (12,2)

bin_supported	We support domestic and international cards. We can configure this in Merchant Settings as 'Domestic', 'International' or 'Both' to specify the supported BINs. Merchants have the ability to override this setting at runtime by passing a request parameter viz. D – Domestic I – International B - Both	Alphabet (1)
trans_fee	Transaction fee applicable for the transaction.	numeric(12,2)
service_tax	Service Tax on fees chargeable to customers (Optional)	Numeric (12,2)

***Disclaimer: "billing_zip" parameter is "Mandatory" for Insurance domain merchants creating SI profile with AMEX card scheme.**

Response Handling:

Please find the below sample of encrypted and decrypted response which gets posted on merchant redirect_url parameter which is passed in the request, once the transaction gets completed. With encResp, orderNo is passed in the response which would be in plain text.

encResp:

a66f9ed977ec2b4a46ef412a180b0f5583afbe7a4db482f37369a2fb68e0ae78b119411a3b11e76535722f8182875213f0145f2ffd6fec56eccec876ea19f0c39bae9f189da5e725fe86f72d4b621ca9e07d30f104bea489b06aa65c501161ac6a188d2aaa24946374bb36f1b780ad60e33188132df818b0054eded3809700f83a7f687b7a402410ff610bab10f8b52286584c46cb8d6ab912abdd2f9ecbc59ca5799a6854489eabd810cc0089a02ab2e6a5188665ca7e421f91c0af04a810a7153880869578b783b4b10a9ccab376e53348f6383e764e05f6b53c6ca77ad4fe778077b06cf82b9e000a00720279d1bdcd48e71a453ba1a160e7e976d41ded68b265adadda42deaed1e212051c72ffd79337b0d5aff02e06c014b7b102e9f27c9447e06be65dbf8b061d3ea651e8648aea5b26a7064f69781c9d3033866baa2515be47329f78bacbe6af740220c0ec8364dafdad210b145b8904895d2c827398ab368c19ea44b5a95f154f752e1875f067d87961aff858b4b8e84bd414912f4f5b6e986568bd245c54a5618c43c5f7e927267d18f16b405560add7328880ad1c234b0f384c1b960f968776157359247559 0acb10a33d4aa08b84e9c389eb6fab0828ef4a7f63a870aa3e43e267e3a58da57b749d41df8d82e6c49779ba5e9ff0acbff436fc72dd8dd1b5cc77acd374506ddb06b83236efe374aea3676516f902316d79e769dd45c01181c28b5f92482b82e265032d8929e59a38b5c577e753b3229bfd2d608ac0ecdbe8910f1e6c495a4d7e4cb79f0a7ff73321d3d46812826a9cfdb1ed2d6ce93afcbdcadc8d0d0238415728b86e4702b72ea0d3bbd7d24a7577b9d578487e040cc6e218e4e7ec0730819a6b9ed72cd1f5697507286d7fd38a035273e2e25207f0ff1a32dd2be67697b09afb7c06adb15c0045831daf516b1b774d50f5c7fba3432f4317917b57138b92c082650aba95e63dd75ccc0a9bba9071389fae6a52897f732576130a52e13d121ce3b9cf1955cc8147d31d593de6613eb6dafa270cb42e8d454dd6c3e076d16ae563cb03606007a7af20be1dd4e7ba44cf4b99423174a4c2f33f9632d92045ff429b1004985009fa2671bc7b9b162b54657b75c76b0c896f4cc11992312373150e8343f58dc5ff77bc80ee850038e1bfe275fec4a33041d04bf8ada8aaff177850

3b3c378b0c7ff36a0ae5348a62c20259a96ce707b377ccd7f1796e2867eaa486b7709ce1b960b01cb10d8baceeaa80
96a58c2e3a820496abb08a8eab359ee113d2ad0cd2058374e8d0e37269525e1a77bb28af42b15760ed7f43888ef7d
2**&orderNo=123654789**

Merchant needs to decrypt the above encrypted response with the help of correct working key which was used in encryption in the request to decrypt the response. Merchant has to rely on **order_status** parameter to get the exact transaction status.

Decrypted Response ::

order_id=123654789&tracking_id=311008298114&bank_ref_no=1652448720867&order_status=Success&failure_message=&payment_mode=Net Banking&card_name=AvenuesTest&status_code=null&status_message=Y&currency=INR&amount=1.00&billing_name=Charli&billing_address=Room no 1101, near Railway station Ambad&billing_city=Indore&billing_state=MP&billing_zip=425001&billing_country=India&billing_tel=9876543210&billing_email=test@test.com&delivery_name=Chaplin&delivery_address=room no.701 near bus stand&delivery_city=Hyderabad&delivery_state=Andhra&delivery_zip=425001&delivery_country=India&delivery_tel=9876543210&merchant_param1=additional Info&merchant_param2=additional Info&merchant_param3=additional Info&merchant_param4=additional Info&merchant_param5=additional Info&vault=N&offer_type=null&offer_code=null&discount_value=0.0&mer_amount=1.00&eci_value=null&retry=N&response_code=0&billing_notes=&trans_date=13/05/2022 19:04:51&bin_country=

Contact Details

For any assistance in integrating CCAvenue payment gateway kindly contact:

CCAvenue Technical Support

Email: alliancesupport@ccavenue.com