

# Integration Document



Version 3.1

# INDEX

|    |   |    |
|----|---|----|
| 1. | <u>CCAvenue Integration Document</u>  | 3  |
| 2. | <u>Testing and Production Environment</u>   | 4  |
| 3. | <u>Integration Methods</u>  | 5  |
| 4. | <u>Processing orders using CCAvenue billing page</u>                                      | 6  |
|    | a. <u>Basic steps involved in integration with the CCAvenue billing page</u>              | 7  |
| 5. | <u>Processing orders using custom checkout form</u>                                       | 22 |
|    | a. <u>Basic steps involved in fetching payment options to create custom checkout form</u> | 23 |
| 6. | <u>Vault feature for storing card</u>   | 32 |
| 7. | <u>Response Parameters</u>  | 34 |
| 8. | <u>Contact Details</u>  | 42 |

# 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** - 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.
3. **CCAvenue "Direct Connect"** - 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.

## **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)   |
| <b>merchant_id</b>  | Merchant Id is a unique identifier generated by CCAvenue for each activated merchant.   | Numeric   |
| <b>order_id</b>     | This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. <b>CCAvenue will not check the uniqueness of this order id.</b> As it generates a unique payment reference number for each order which is sent by the merchant.   | Alphanumeric (30)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z),<br>Numbers,- (hyphen),<br>/ (slash), _ (underscore) |
| <b>currency</b>     | Currency in which you want to process the transaction.<br>INR – Indian Rupee<br>USD – United States Dollar<br>SGD – Singapore Dollar<br>GBP – Pound Sterling<br>EUR – Euro, official currency of Eurozone   | Alphabets (3)   |
| <b>amount</b>       | Order amount  | Numeric (12, 2)   |
| <b>redirect_url</b> | 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)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z),<br>Numbers, / (slash), _ (underscore)              |
| <b>cancel_url</b>   | CCAvenue will redirect the customer to this URL if the customer cancels the transaction on the billing page.  | Alphanumeric (100)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z),<br>Numbers, / (slash), _ (underscore)              |
| <b>language</b>     | 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.

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

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

|                        |  |   |
|------------------------|--|---|
| <b>delivery_tel</b>    | Shipping phone number  | Numeric (20)  |
| <b>merchant_param1</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.<br><b>NOTE:</b> This parameter value would get displayed on the payment page. | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param2</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param3</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param4</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param5</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (5000)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen) |
| <b>promo_code</b>      | 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)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers  |

|            |  |   |
|------------|--|---|
| <b>tid</b> | 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. <b><i>The uniqueness of TID is valid for 24 hours only.</i></b> | Numeric(17)<br><br><b>Characters allowed:</b><br>Only numbers |
|------------|--|---|

**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](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=AVGPUUZ24BB27BBBB](https://secure.ccavenue.com/transaction/transaction.do?command=initiateTransaction&encRequest=ccee6b03e00fa4f3657cf56a57bc681341cede3be1c5a077510f239511efa413dfd5ac3dacdbe0ddd2ae6806fc9c04229be8b4535896b0cab939ccd4ec9a9560&access_code=AVGPUUZ24BB27BBBB)

## Request parameters for Standing Instruction Information

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

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

|                         |   |          |
|-------------------------|---|----------|
| <b>si_frequency_no</b>  | <p>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.</p> <p>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.</p>   | Numeric  |
| <b>si_billing_cycle</b> | <p>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.</p> <p>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.</p> | Numeric  |
| <b>si_start_date</b>    | <p>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.</p>  | datetime |

## **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: 'json',
        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.

| <b>Required Parameters</b> |  |  |
|----------------------------|--|--|
| <b>Name</b>                | <b>Description</b>   | <b>Type (length)</b>   |
| <b>merchant_id</b>         | Merchant Id is a unique identifier generated by CCAvenue for each activated merchant.  | Numeric  |
| <b>order_id</b>            | This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. <b>CAvenue will not check the uniqueness of this order id.</b> As it generates a unique payment reference number for each order which is sent by the merchant.   | Alphanumeric (30)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z),<br>Numbers,- (hyphen),<br>/ (slash), ,_ (underscore) |
| <b>currency</b>            | Currency in which you want to process the transaction.<br>INR – Indian Rupee<br>USD – United States Dollar<br>SGD – Singapore Dollar<br>GBP – Pound Sterling<br>EUR – Euro, official currency of Eurozone  | Alphabets (3)  |
| <b>amount</b>              | Order amount   | Numeric (12, 2)  |
| <b>redirect_url</b>        | CAvenue 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)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z),<br>Numbers, / (slash),_<br>(underscore)             |
| <b>cancel_url</b>          | CAvenue will redirect the customer to this URL if the customer cancels the transaction on the billing page.  | Alphanumeric (100)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z),<br>Numbers, / (slash),_<br>(underscore)             |

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

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

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

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

|                        |  |   |
|------------------------|--|---|
| <b>merchant_param1</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.<br><b>NOTE:</b> This parameter value would get displayed on the payment page. | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param2</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param3</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param4</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param5</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (5000)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (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.

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

|                         |   |          |
|-------------------------|---|----------|
| <b>si_frequency_no</b>  | <p>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.</p> <p>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.</p>   | Numeric  |
| <b>si_billing_cycle</b> | <p>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.</p> <p>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.</p> | Numeric  |
| <b>si_start_date</b>    | <p>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.</p>  | datetime |



## 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.

| <b>Vault Information</b>   |  |                                       |    |
|----------------------------|--|---------------------------------------|----|
| <b>customer_identifier</b> | The identifier against which the card information is to be stored or retrieved<br>Email ID<br>Customer ID<br>Mobile number | Alphanumeric, '@' and '.' are allowed | 70 |

## **Processing orders using CCAvenue Direct Connect**

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: 'json',
        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.

| <b>Required Parameters</b> |  |   |
|----------------------------|--|---|
| <b>Name</b>                | <b>Description</b>   | <b>Type (length)</b>  |
| <b>merchant_id</b>         | Merchant Id is a unique identifier generated by CCAvenue for each activated merchant.  | Numeric   |
| <b>order_id</b>            | This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. <b>CAvenue will not check the uniqueness of this order id.</b> As it generates a unique payment reference number for each order which is sent by the merchant.   | Alphanumeric (30)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z),<br>Numbers,- (hyphen),<br>/ (slash), _ (underscore) |
| <b>currency</b>            | Currency in which you want to process the transaction.<br>INR – Indian Rupee<br>USD – United States Dollar<br>SGD – Singapore Dollar<br>GBP – Pound Sterling<br>EUR – Euro, official currency of Eurozone  | Alphabets (3)   |
| <b>amount</b>              | Order amount   | Numeric (12, 2)   |
| <b>redirect_url</b>        | CAvenue 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)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z),<br>Numbers, / (slash), _ (underscore)              |
| <b>cancel_url</b>          | CAvenue will redirect the customer to this URL if the customer cancels the transaction on the billing page.  | Alphanumeric (100)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z),<br>Numbers, / (slash), _ (underscore)              |

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

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

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



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

|                         |  |   |
|-------------------------|--|---|
| <b>device_parameter</b> | 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)<br><b>Characters allowed:</b><br>MOB<br>PC  |
| <b>merchant_param1</b>  | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.<br><b>NOTE:</b> This parameter value would get displayed on the payment page. | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param2</b>  | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param3</b>  | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param4</b>  | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen)  |
| <b>merchant_param5</b>  | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (5000)<br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma, circular brackets, /(slash), dot, - (hyphen) |

|   |  |  |
|---|--|--|
| <b>sub_account_id</b><br><b>(Mandatory for sub_account_id integration only)</b> | 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.

|                              |   |  |
|------------------------------|---|--|
| <b>si_type</b><br>(required) | This parameter is used to identify whether the standing instruction request is for the fixed amount or for variable amount. Expected values:<br><br>Fixed<br><br>Variable   | Alphabet(8)  |
| <b>si_mer_ref_no</b>         | This parameter can be used by the merchant to send a unique identifier. E.G. For insurance – Policy number.<br><br>It can also be a customer reference number.  | Alphanumeric (30)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z),<br>Numbers,- (hyphen),<br>/ (slash), ,_ (underscore) |
| <b>si_amount</b>             | 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)   |
| <b>si_setup_amount</b>       | This is a mandatory field and is required as part of the SI creation process. This is a one-time charge   |  |
| <b>si_frequency</b>          | This will be required only in case of “Fixed” type standing instruction. Expected values:<br><br>Week<br><br>Month<br><br>Year<br><br>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)  |

|                         |   |          |
|-------------------------|---|----------|
| <b>si_frequency_no</b>  | <p>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.</p> <p>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.</p>   | Numeric  |
| <b>si_billing_cycle</b> | <p>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.</p> <p>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.</p> | Numeric  |
| <b>si_start_date</b>    | <p>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.</p>  | datetime |

## Response Parameters

CCAvenue PG will return following parameters:

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

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

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



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

|                        |  |  |
|------------------------|--|--|
| <b>merchant_param3</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br># (hash), Comma,<br>circular brackets,<br>/(slash), dot, - (hyphen) |
| <b>merchant_param4</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (500)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br>  |
| <b>merchant_param5</b> | This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.   | Alphanumeric (5000)<br><br><b>Characters allowed:</b><br>Alphabet (A-Z), (a-z).<br>Numbers<br>   |
| <b>vault</b>           | 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)<br><br><b>Characters allowed:</b><br>Y or N   |
| <b>offer_type</b>      | This parameter can be used for sending additional information if customer has used any discount or promotion while completing the transaction.<br>If customer is using discount-coupon, value of this parameter would be discount.<br>If customer is using promo-code, value of this parameter would be promotion. | Alphabets (9)  |

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

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

|                      |   |                |
|----------------------|---|----------------|
| <b>bin_supported</b> | 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.<br><br>D – Domestic<br><br>I – International<br><br>B - Both | Alphabet (1)   |
| <b>trans_fee</b>     | Transaction fee applicable for the transaction.   | numeric(12,2)  |
| <b>service_tax</b>   | Service Tax on fees chargeable to customers (Optional)  | Numeric (12,2) |

## 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:

```
a66f9ed977ec2b4a46ef412a180b0f5583afbe7a4db482f37369a2fb68e0ae78b119411a3b11e76535722f81828752
13f0145f2ffd6fec56ecce876ea19f0c39bae9f189da5e725fe86f72d4b621ca9e07d30f104bea489b06aa65c501161
ac6a188d2aaa24946374bb36f1b780ad60e33188132df818b0054eded3809700f83a7f687b7a402410ff610bab10f8
b52286584c46cb8d6ab912abdd2f9ecbc59ca5799a6854489eabd810cc0089a02ab2e6a5188665ca7e421f91c0af04
a810a7153880869578b783b4b10a9ccab376e53348f6383e764e05f6b53c6ca77ad4fe778077b06cf82b9e000a007
20279d1bdcd48e71a453ba1a160e7e976d41ded68b265adadda42deaed1e212051c72ffd79337b0d5aff02e06c014
b7b102e9f27c9447e06be65dbf8b061d3ea651e8648aea5b26a7064f69781c9d3033866baa2515be47329f78bacbe
6af740220c0ec8364dafdad210b145b8904895d2c827398ab368c19ea44b5a95f154f752e1875f067d87961aff858b
4b8e84bd414912f4f5b6e986568bd245c54a5618c43c5f7e927267d18f16b405560add7328880ad1c234b0f384c1b
960f9687761573592475590acb10a33d4aa08b84e9c389eb6fab0828ef4a7f63a870aa3e43e267e3a58da57b749d4
1df8d82e6c49779ba5e9ff0acbff436fc72dd8dd1b5cc77acd374506ddb06b83236efe374aea3676516f902316d79e7
69dd45c01181c28b5f92482b82e265032d8929e59a38b5c577e753b3229bfd2d608ac0ecdbe8910f1e6c495a4d7e4
cb79f0a7ff73321d3d46812826a9cfdb1ed2d6ce93afcbdcadc8d0d0238415728b86e4702b72ea0d3bbd7d24a7577
b9d578487e040cc6e218e4e7ec0730819a6b9ed72cd1f5697507286d7fd38a035273e2e25207f0ff1a32dd2be6769
7b09afb7c06adb15c0045831daf516b1b774d50f5c7fba3432f4317917b57138b92c082650aba95e63dd75ccc0a9bb
a9071389fae6a52897f732576130a52e13d121ce3b9cf1955cc8147d31d593de6613eb6dafa270cb42e8d454dd6c3
e076d16ae563cb03606007a7af20be1dd4e7ba44cf4b99423174a4c2f33f9632d92045ff429b1004985009fa2671bc
7b9b162b54657b75c76b0c896f4cc11992312373150e8343f58dc5ff77bc80ee850038e1bfe275fec4a33041d04bf8a
da8aaff1778503b3c378b0c7ff36a0ae5348a62c20259a96ce707b377ccd7f1796e2867eaa486b7709ce1b960b01cb
10d8baceaaa8096a58c2e3a820496abb08a8eab359ee113d2ad0cd2058374e8d0e37269525e1a77bb28af42b1576
```

Oed7f43888ef7d2&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](mailto:alliancesupport@ccavenue.com)