

# **CCAvenue Integration Document**



**Version 2.9**

**Phone: +91 022-67425555**

**April 2018**

## 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 CCAvenue iFrame Checkout</u>	14
a. <u>Basic steps involved in integration CCAvenue iFrame into Checkout page</u>	15
6. <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
7. <u>Vault feature for storing card</u>	32
8. <u>Processing orders using CCAvenue shopping cart</u>	33
a. <u>Basic steps involved in integration with the CCAvenue shopping cart</u>	34
9. <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>

CCAvenue production URL is: <https://secure.ccavenue.com>

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. **CCAvenue iFrame Checkout** - Fastest and easiest way to enable payments on your website. CCAvenue iframe checkout is a pre-configured form, which validates the payment data, allows user to store card information to expedite the payment process in future. CCAvenue iFrame checkout also handles PCI compliance.
3. **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.
4. **CCAvenue shopping cart** - CCAvenue provides merchants with a product management module and a customizable shopping cart thereby eliminating the need for developing/maintaining their own.
5. **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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), _ (underscore)
<b>currency</b>	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)
<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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), 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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), 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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>billing_address</b>	Customer's billing address	Alphanumeric (150)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, / (slash), dot, - (hyphen) Space in between words.
<b>billing_city</b>	Customer's billing city	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>billing_state</b>	Customer's billing state	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>billing_zip</b>	Customer's billing zip code	Alphanumeric (15)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers
<b>billing_country</b>	Customer's billing country	Alphabets (50)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>billing_tel</b>	Customer's phone number	Numeric (20)

<b>billing_email</b>	Customer's email address	Alphanumeric (70)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers @ (at), dot, _ (underscore)
<b>delivery_name</b>	Recipient's name	Alphabets (60)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_address</b>	Shipping address	Alphanumeric (150)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
<b>delivery_city</b>	Shipping city	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_state</b>	Shipping state	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_zip</b>	Shipping zip code	Alphanumeric (15)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers
<b>delivery_country</b>	Shipping country	Alphabets (50)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). 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.	Alphanumeric (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers

<b>tid</b>	<p>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></p>	<p>Numeric(17)</p> <p><b>Characters allowed:</b> Only numbers</p>
------------	---	---

## 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> (required)	<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>	<p>Alphanumeric (30)</p> <p><b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), , _ (underscore)</p>
<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 CCAvenue iFrame Checkout**

This is the fastest and the easiest way to enable payments on merchant website. CCAvenue iframe checkout will enable the merchants to display the payment options on their checkout page and there by collect the payment information on their checkout page.

### **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 billing page customer selects the payment option and enters the payment information in the CCAvenue iFrame which is loaded after submitting the order information like merchant id, amount, currency, shipping information (optional) and billing information (optional).
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 integration CCAvenue iFrame into Checkout 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.

### Sample JSP Code

```
<div id="paymentDiv"></div>
<iframe width="482" height="500" scrolling="No" frameborder="0" id="paymentFrame" src=""></iframe>

<script type="text/javascript">
    $(document).ready(function() {
        $(function() {
            $("#checkout").live('click', function(e) {
                $("#paymentDiv").children().remove();
                var formData = $("form[name='customerData']").serialize();
                $.post("/transaction/jsp/iframe/iframeEncReq.jsp?", formData, function(data) {
                    var encRequest, Merchant_Id, url;
                    $("#paymentDiv").append(data);
                    Merchant_Id = $("#paymentDiv").find("#merchantId").val();
                    encRequest = $("#paymentDiv").find("#encRequest").val();
                    url = "http://secure.ccavenue.com/transaction/transaction.do?command=initiateTransaction&Merchant_Id="
                        + Merchant_Id + "&encRequest=" + encRequest;
                    $("#paymentFrame").attr("src", url);
                });
                e.preventDefault();
            });
        });

        $('iframe#paymentFrame').load(function() {
            window.addEventListener('message', function(e) {
                $("#paymentFrame").css("height", e.data['newHeight'] + 'px');
            }, false);
        });
    });
</script>
```

### Request Parameters



Merchant must send the following parameters to the CCAvenue PG for initiating the transaction and loading the CCAvenue iFrame.

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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), _ (underscore)
<b>currency</b>	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)
<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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), 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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers, / (slash), _ (underscore)
<b>integration_type</b>	Describes the type of iFrame flow iframe_normal – the bank pages will be displayed in the same tab as the payments page.	Exact value expected iframe_normal

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

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

<b>delivery_country</b>	Shipping country	Alphabets (50)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_tel</b>	Shipping phone number	Numeric (22)  <b>Characters allowed:</b> Numbers and - (Hyphen)
<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.	Alphanumeric (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers

## 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> (required)	<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>	<p>Alphanumeric (30)</p> <p><b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)</p>
<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: '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)
<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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), _ (underscore)
<b>currency</b>	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)
<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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), 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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers, / (slash), _ (underscore)

<b>payment_option</b>	Payment option selected by the customer OPTCRDC - Credit Card OPTDBCRD - Debit Card OPTNBK - Net Banking OPTCASHC - Cash Card OPTMOBP - Mobile Payments	Alphabets (10)
<b>card_type</b>	Type of card used by the customer.  CRDC - Credit Card DBCRD - Debit Card NBK - Net Banking CASHC - Cash Card 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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z).
<b>data_accept</b>	Resend the parameter value received at the time of fetching the payment options. <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)  <b>Characters allowed:</b> 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.

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

<b>billing_email</b>	Customer's email address	Alphanumeric (70)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers @ (at), dot, _ (underscore)
<b>delivery_name</b>	Recipient's name	Alphabets (50) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_address</b>	Shipping address	Alphanumeric (150)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, / (slash), dot, - (hyphen) Space in between words.
<b>delivery_city</b>	Shipping city	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_state</b>	Shipping state	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_zip</b>	Shipping zip code	Alphanumeric (15)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers
<b>delivery_country</b>	Shipping country	Alphabets (50)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). 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.	Alphanumeric (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> 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.

<b>si_type</b> (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)
<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.  It can also be a customer reference number.	Alphanumeric (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (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:  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)



<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.

Vault Information			
<b>customer_identifier</b>	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**

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.

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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers, - (hyphen), / (slash), _ (underscore)
<b>currency</b>	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)
<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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), 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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers, / (slash), _ (underscore)

<b>payment_option</b>	Payment option selected by the customer OPTCRDC - Credit Card OPTDBCRD - Debit Card OPTNBK - Net Banking OPTCASHC - Cash Card OPTMOBP - Mobile Payments	Alphabets (10)
<b>card_type</b>	Type of card used by the customer.  CRDC - Credit Card DBCRD - Debit Card NBK - Net Banking CASHC - Cash Card 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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z).
<b>data_accept</b>	Resend the parameter value received at the time of fetching the payment options. <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)  <b>Characters allowed:</b> 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.

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

<b>billing_email</b>	Customer's email address	Alphanumeric (70)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers @ (at), dot, _ (underscore)
<b>delivery_name</b>	Recipient's name	Alphabets (50) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_address</b>	Shipping address	Alphanumeric (150)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, / (slash), dot, - (hyphen) Space in between words.
<b>delivery_city</b>	Shipping city	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_state</b>	Shipping state	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_zip</b>	Shipping zip code	Alphanumeric (15)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers
<b>delivery_country</b>	Shipping country	Alphabets (50)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). 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)  <b>Characters allowed:</b> MOB 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.	Alphanumeric (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> 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.

<b>si_type</b> (required)	<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>	<p>Alphanumeric (30)</p> <p><b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)</p>
<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 CCAvenue shopping cart**

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

### **Process flow**

1. Customer views the product/service displayed on your website.
2. He selects a product by clicking on the 'add to cart' button.
3. The Customer is redirected to the CCAvenue shopping cart page where the product added is displayed.
4. On the shopping card the customer can select variants, extras and update the quantity of the product he has added.
5. The customer may opt to continue shopping and he will be taken back to your website.
6. If the customer opts to checkout he will be taken to the CCAvenue billing page where billing, shipping and payment information is entered by the customer.
7. On submission of the transaction information, CCAvenue initiates the authorization process by connecting to the relevant bank/processing organization.
8. 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 shopping cart:

**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.

```
<ahref="https://test.ccavenue.com
/transaction/txn/shopcart/access_code,product_id,currency,shopping_url/language">Buy
Now</a>
```

```
<ahref="https://test.ccavenue.com/transaction/txn/shopcart/EUAF9DAAJWHDGFY3,546,INR,htt
p://yoursite.com/shop.htm/EN"> Add To Cart </a>
```

```
<ahref="https://test.ccavenue.com/transaction/txn/shopcart/EUAF9DAAJWHDGFY3,546,INR,htt
p://yoursite.com/shop.htm/EN"><img src='add2cart.jpg'></a>
```

## Request Parameters

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>product_id</b>	This ID is used to identify the product. This is available in your CCAvenue MARS panel.	Alphanumeric (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers, - (hyphen), / (slash), , _ (underscore)
<b>currency</b>	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)
<b>shopping_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)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers, / (slash), _ (underscore)
<b>language</b>	CCAvenue billing page is multi-lingual. Currently we are displaying the page in English (Code - EN).	Alphabet(5)

The currency and language sent in the first request will be set for the session of the shopping cart.



## 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	Alphanumeric (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers, # (hash), /(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. Success Failure Aborted Invalid	Alphabets (15)
<b>failure_message</b>	Reason for failure.	Alphanumeric
<b>payment_mode</b>	The payment mode used by customer  IVRS EMI Credit Card Net banking Debit Card Cash Card UPI 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>	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	Alphabets (3)
<b>Amount</b>	Order amount	Numeric (12, 2)
<b>billing_name</b>	Name of the customer	Alphabets (60)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>billing_address</b>	Customer's billing address	Alphanumeric (150)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
<b>billing_city</b>	Customer's billing city	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>billing_state</b>	Customer's billing state	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.

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

<b>delivery_city</b>	Shipping city	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_state</b>	Shipping state	Alphabets (30)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Space in between words.
<b>delivery_zip</b>	Shipping zip code	Alphanumeric (15)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers
<b>delivery_country</b>	Shipping country	Alphabets (50)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). 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.	Alphanumeric (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (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 (100)  <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
<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)  <b>Characters allowed:</b> 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. 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)

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

<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 <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) 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>	<p>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.</p> <p>D – Domestic</p> <p>I – International</p> <p>B - Both</p>	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)





## Contact Details

For any assistance in integrating CCAvenue payment gateway kindly contact:

### CCAvenue Technical Support

**Tel** : +91 022 67425555 **Ext:** 401-409 / 421-423

**Email:** [service@ccavenue.com](mailto:service@ccavenue.com)