KOTAK Integration Document

Version 2.7

Phone: +91 022-67425555

February 2017

INDEX

KOTAK Integration Document

Introduction

KOTAK 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 KOTAK 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

KOTAK test and production environments are separate.

Merchants need an active KOTAK account to use the test environment and production environment. Merchants will have to log in to their KOTAK 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.

KOTAK test URL is: https://test.ccavenue.com

KOTAK production URL is: https://secure.ccavenue.com

To test the integration login to your Kotak 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

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

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

2. **KOTAK iFrame Checkout -** Fastest and easiest way to enable payments on your website. KOTAK 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. KOTAK 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 Kotak server for payment processing. KOTAK can also store the payment information of the customer to expedite the payment process in future.

4. **KOTAK shopping cart -** KOTAK provides merchants with a product management module and a customizable shopping cart thereby eliminating the need for developing/maintaining their own.

5. **KOTAK "Direct Connect" -** This integration enables you to deliver payment services directly through your website without redirecting your users to KOTAK. 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.

KOTAK billing page (Non-Seamless)

Processing orders using KOTAK billing page

KOTAK billing page helps you avoid the hassle of developing and managing your own billing page. KOTAK 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 KOTAK billing page where billing, shipping and payment information is entered by the customer.

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

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

Basic steps involved in integration with the KOTAK 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 KOTAK PG for processing an order.

Required Parameters

Name	**Description**	**Type (length)**
merchant_id	Merchant Id is a unique identifier generated by KOTAK for each activated merchant.	Numeric
order_id	This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. KOTAK will not check the uniqueness of this order id as it generates a unique payment reference number for each order which is sent by the merchant.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
currency	Currency in which you want to process the transaction. INR – Indian Rupee USD – United States Dollar SGD – Singapore Dollar GBP – Pound Sterling EUR – Euro, official currency of Eurozone	Alphabets (3)
amount	Order amount	Numeric (12, 2)
redirect_url	KOTAK 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 KOTAK confirmation page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
cancel_url	KOTAK will redirect the customer to this URL if the customer cancels the transaction on the billing page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
language	KOTAK billing page is multi-lingual. Currently we are displaying the page in English (Code - EN).	Alphabet(5)

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

Billing and Shipping Information

Name	Description	Type (length)
billing_name	Name of the customer	Alphabets (60) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_address	Customer’s billing address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
billing_city	Customer’s billing city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_state	Customer’s billing state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_zip	Customer’s billing zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
billing_country	Customer’s billing country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_tel	Customer’s phone number	Numeric (20)
billing_email	Customer’s email address	Alphanumeric (70)

		Characters allowed: Alphabet (A-Z), (a-z). Numbers @ (at), dot,_ (underscore)
delivery_name	Recipient’s name	Alphabets (60) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_address	Shipping address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
delivery_city	Shipping city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_state	Shipping state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_zip	Shipping zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
delivery_country	Shipping country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_tel	Shipping phone number	Numeric (20)
merchant_param1	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z).

		Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param2	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param3	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param4	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param5	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
promo_code	This parameter is used for sending the code of the promotion you have created in the KOTAK MARs by which you may offer specific discounts to customers using specific payment options.	Alphanumeric (20) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
tid	This parameter is used for sending the unique identifier to identify uniqueness of the order. This is an optional parameter. Value for this parameter can be generated using the piece of code given in the integration kit.	Numeric(17) **Characters allowed:** Only numbers

Request parameters for Standing Instruction Information

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

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

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

Processing orders using KOTAK iFrame Checkout

This is the fastest and the easiest way to enable payments on merchant website. KOTAK 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 KOTAK 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, KOTAK initiates the authorization process by connecting to the relevant bank/processing organization.

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

Basic steps involved in integration KOTAK iFrame into Checkout page:

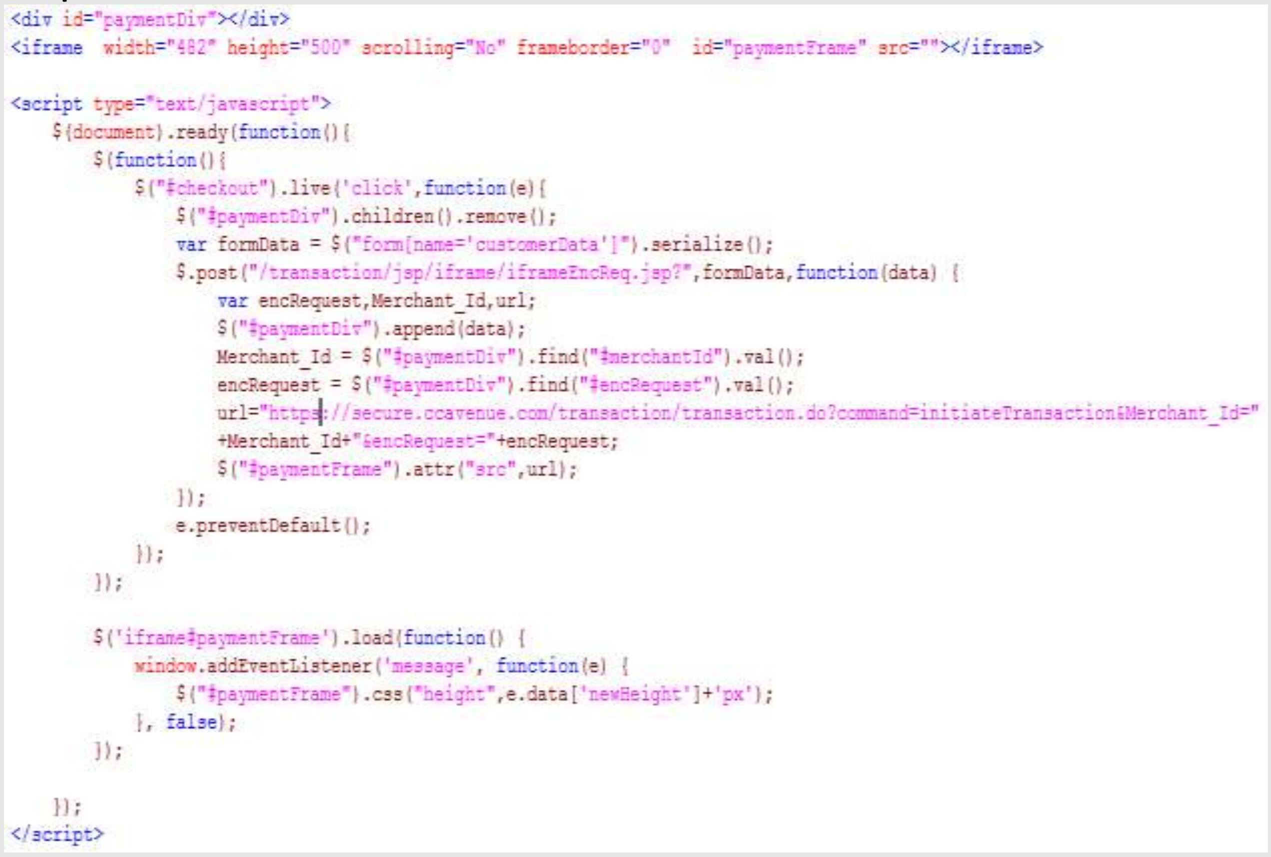
Set Up: Download the CCAvenue client library from the MARS panel. Click on “Resources” on the navigation bar of the Dashboard and click “Integration Kit”. You will have to use the CCAvenue transaction file (e.g. ccavRequestHandler.php) to initiate the payment process.

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 KOTAK 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="https://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 KOTAK PG for initiating the transaction and loading the KOTAK iFrame.

Required Parameters

Name	Description	Type (length)
merchant_id	Merchant Id is a unique identifier generated by KOTAK for each activated merchant.	Numeric
order_id	This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. KOTAK will not check the uniqueness of this order id as it generates a unique payment reference number for each order which is sent by the merchant.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
currency	Currency in which you want to process the transaction. INR – Indian Rupee USD – United States Dollar SGD – Singapore Dollar GBP – Pound Sterling EUR – Euro, official currency of Eurozone	Alphabets (3)
amount	Order amount	Numeric (12, 2)
redirect_url	KOTAK 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 KOTAK confirmation page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
cancel_url	KOTAK will redirect the customer to this URL if the customer cancels the transaction on the billing page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
integration_type	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
language	KOTAK billing page is multi-lingual. Currently we are displaying the page in English (Code - EN).	Alphabet(5)

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

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

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

delivery_country	Shipping country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_tel	Shipping phone number	Numeric (22) **Characters allowed:** Numbers and - (Hyphen)
merchant_param1	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, -(hyphen)
merchant_param2	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, -(hyphen)
merchant_param3	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, -(hyphen)

merchant_param4	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param5	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
promo_code	This parameter is used for sending the code of the promotion you have created in the KOTAK MARs by which you may offer specific discounts to customers using specific payment options.	Alphanumeric (20) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers

Request parameters for Standing Instruction Information

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

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

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

Processing orders using custom checkout form

Merchants can build a custom checkout form to collect order and payment information and pass the same to KOTAK 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 KOTAK as a JSON object. Customer enters the payment information and submits the form.

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

4. On receiving the authorization status from the bank, KOTAK 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 KOTAK 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 KOTAK 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 KOTAK transaction file (e.g. ccavRequestHandler.jsp) to initiate the payment process. KOTAK transaction file on receiving the order related data will encrypt the data and forward the encrypted request to the KOTAK 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.

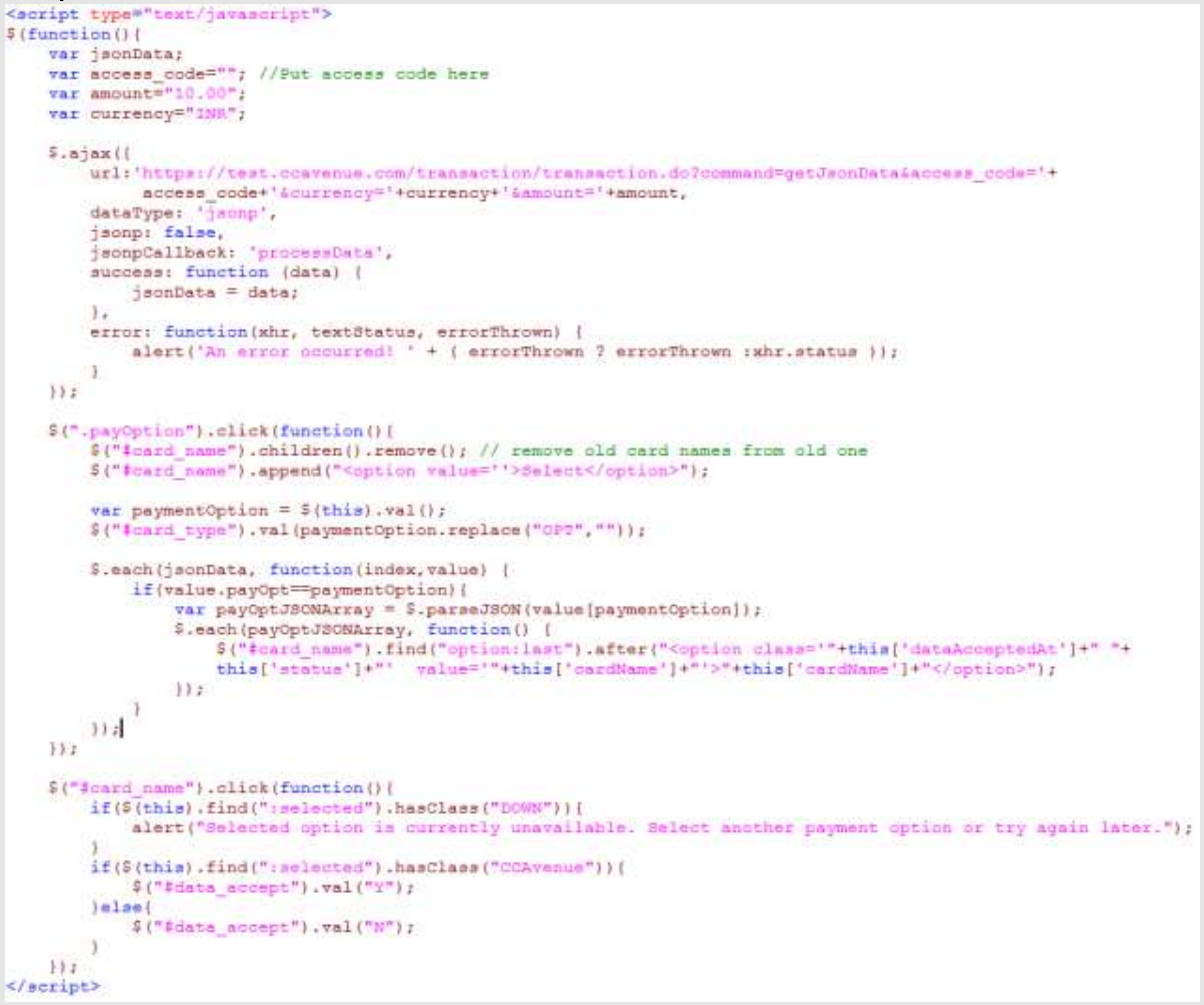
3. **Card Name –** Will contain name of card. E.g. Visa, MasterCard, American Express or and bank name in case of Net banking.

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 KOTAK transaction file (e.g. ccavRequestHandler.jsp) to initiate the payment process. KOTAK transaction file on receiving the order related data will encrypt the data and forward the encrypted request to the KOTAK 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 KOTAK PG for processing an order.

Required Parameters

Name	Description	Type (length)
merchant_id	Merchant Id is a unique identifier generated by KOTAK for each activated merchant.	Numeric
order_id	This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. KOTAK will not check the uniqueness of this order id as it generates a unique payment reference number for each order which is sent by the merchant.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
currency	Currency in which you want to process the transaction. INR – Indian Rupee USD – United States Dollar SGD – Singapore Dollar GBP – Pound Sterling EUR – Euro, official currency of Eurozone	Alphabets (3)
amount	Order amount	Numeric (12, 2)
redirect_url	KOTAK 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 KOTAK confirmation page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
cancel_url	KOTAK will redirect the customer to this URL if the customer cancels the transaction on the billing page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
payment_option	Payment option selected by the customer OPTCRDC - Credit Card OPTDBCRD - Debit Card OPTNBK - Net Banking OPTCASHC - Cash Card OPTMOBP - Mobile Payments	Alphabets (10)

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

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

Name	**Description**	**Type (length)**
billing_name	Name of the customer	Alphabets (60) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.

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

delivery_address	Shipping address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
delivery_city	Shipping city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_state	Shipping state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_zip	Shipping zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
delivery_country	Shipping country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_tel	Shipping phone number	Numeric (20)
merchant_param1	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)

merchant_param2	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param3	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param4	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param5	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)

Request parameters for Standing Instruction Information

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

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

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

Vault feature for storing card

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

KOTAK 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. KOTAK PG will store the card information against the customer identifier.

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

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

Processing orders using KOTAK Direct Connect

This integration will enable you to deliver payment services directly through your website without redirecting your users to KOTAK. 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 KOTAK 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 KOTAK to fetch the request payload for the payment option selected by the user.

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

5. KOTAK 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 KOTAK 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 KOTAK 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 KOTAK transaction file (e.g. ccavRequestHandler.jsp) to initiate the payment process. KOTAK transaction file on receiving the order related data will encrypt the data and forward the encrypted request to the KOTAK 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.

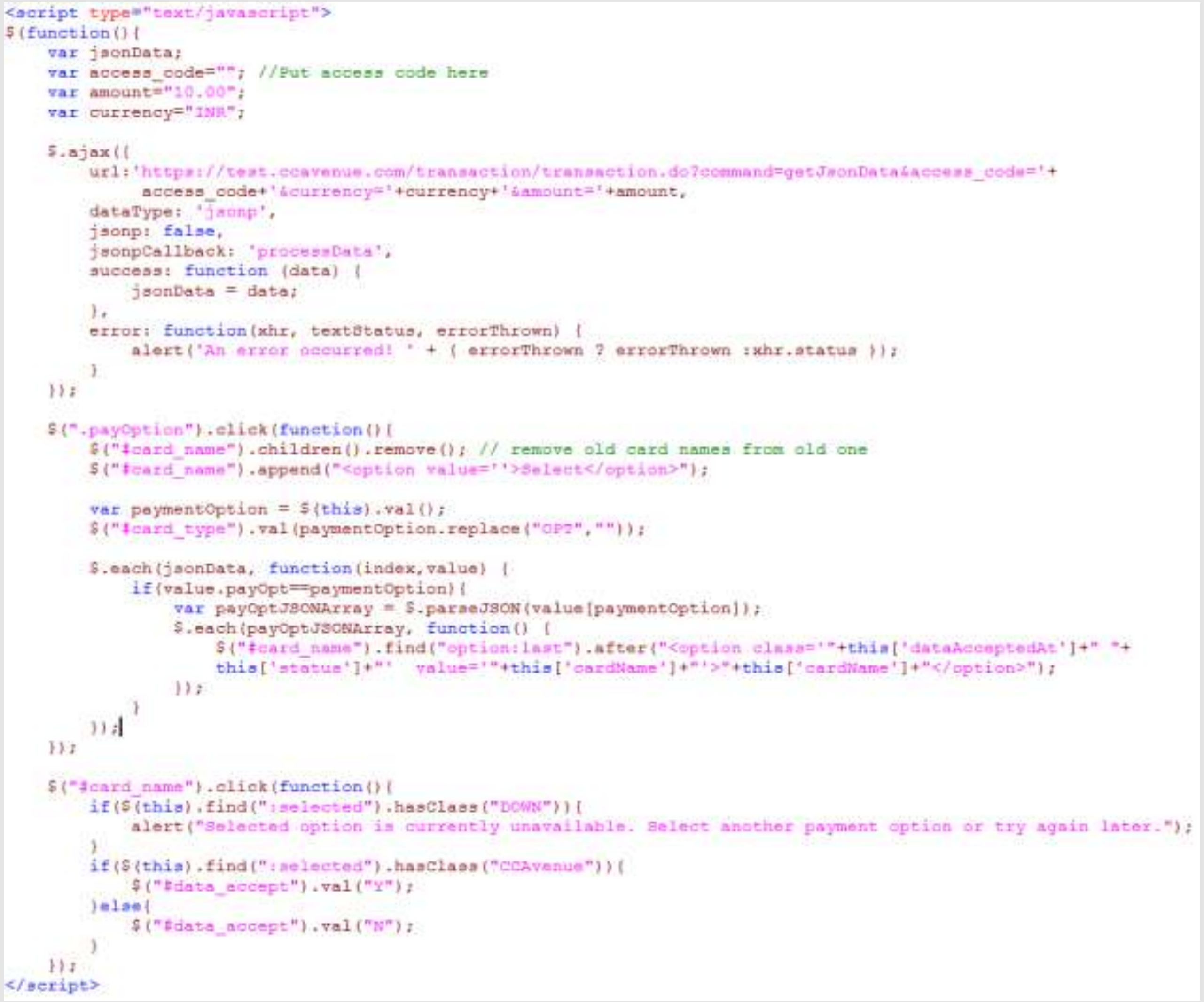
3. **Card Name –** Will contain name of card. E.g. Visa, MasterCard, American Express or and bank name in case of Net banking.

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

Sample Code



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

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

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

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

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

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

Sample code for handling Payload at your end:

For HTML

```
HttpClient vClient = new HttpClient();
String vResponse =
vClient.processUrlConnectionReq("encRequest="+encRequest+"&access_code="+access_code,"https://te
st.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://te
st.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 KOTAK PG for processing an order.

Required Parameters		
Name	**Description**	**Type (length)**
merchant_id	Merchant Id is a unique identifier generated by KOTAK for each activated merchant.	Numeric
order_id	This ID is used by merchants to identify the order. Ensure that you send a unique id with each request. KOTAK will not check the uniqueness of this order id as it generates a unique payment reference number for each order which is sent by the merchant.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
currency	Currency in which you want to process the transaction. INR – Indian Rupee USD – United States Dollar SGD – Singapore Dollar GBP – Pound Sterling EUR – Euro, official currency of Eurozone	Alphabets (3)
amount	Order amount	Numeric (12, 2)
redirect_url	KOTAK 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 KOTAK confirmation page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
cancel_url	KOTAK will redirect the customer to this URL if the customer cancels the transaction on the billing page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
payment_option	Payment option selected by the customer OPTCRDC - Credit Card OPTDBCRD - Debit Card OPTNBK - Net Banking OPTCASHC - Cash Card OPTMOBP - Mobile Payments	Alphabets (10)

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

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

Name	**Description**	**Type (length)**

billing_name	Name of the customer	Alphabets (60) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_address	Customer's billing address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
billing_city	Customer's billing city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_state	Customer's billing state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_zip	Customer's billing zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
billing_country	Customer's billing country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
billing_tel	Customer's phone number	Numeric (20)
billing_email	Customer's email address	Alphanumeric (70) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers @ (at), dot,_ (underscore)

delivery_name	Recipient's name	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_address	Shipping address	Alphanumeric (150) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words.
delivery_city	Shipping city	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_state	Shipping state	Alphabets (30) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_zip	Shipping zip code	Alphanumeric (15) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers
delivery_country	Shipping country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_tel	Shipping phone number	Numeric (20)
device_parameter	This optional parameter is used only in case Direct Connect integration for in which merchant sends a device type though which transaction is processed.	Alphabets (3) **Characters allowed:** MOB PC

merchant_param1	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param2	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param3	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param4	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param5	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)

Request parameters for Standing Instruction Information

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

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

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

Processing orders using KOTAK shopping cart

KOTAK shopping cart helps you avoid the hassle of developing and managing your own shopping cart. KOTAK 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 KOTAK 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 KOTAK billing page where billing, shipping and payment information is entered by the customer.

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

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

Basic steps involved in integration with the KOTAK shopping cart:

Set Up: Download the KOTAK 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 KOTAK 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 KOTAK transaction file (e.g. ccavRequestHandler.jsp) to initiate the payment process. KOTAK transaction file on receiving the order related data will encrypt the data and forward the encrypted request to the KOTAK 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)**
merchant_id	Merchant Id is a unique identifier generated by KOTAK for each activated merchant.	Numeric
product_id	This ID is used to identify the product. This is available in your KOTAK MARS panel.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers,- (hyphen), / (slash), ,_ (underscore)
currency	Currency in which you want to process the transaction. INR – Indian Rupee USD – United States Dollar SGD – Singapore Dollar GBP – Pound Sterling EUR – Euro, official currency of Eurozone	Alphabets (3)
shopping_url	KOTAK 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 KOTAK confirmation page.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, / (slash),_ (underscore)
language	KOTAK 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

KOTAK PG will return following parameters:

Name	Description	Type (length)
order_id	Unique ID sent by the merchant at the time of initiating the transaction.	Alphanumeric (30) **Characters allowed:** Alphabet (A-Z), (a-z), Numbers, # (hash), /(slash, - (hyphen)
tracking_id	Unique payment reference number generated by KOTAK for each order.	Numeric (12)
bank_ref_no	Reference number generated by the bank for the transaction.	Alphanumeric
order_status	Status of the order. Success Failure Aborted Invalid	Alphabets (1)
failure_message	Reason for failure.	Alphanumeric
payment_mode	The payment mode for the transaction for eg , web , ivrs, emi, netbanking, debit card .	Alphabets
card_name	Specifies the type of credit card, debit card, netbanking etc .	Alphabets
status_code	The status code for this transaction	Numeric (3)
status_message	The status message for this transaction.	Alphanumeric (150)
currency	Currency code in which the transaction was processed. INR – Indian Rupee USD – United States Dollar SGD – Singapore Dollar GBP – Pound Sterling EUR – Euro, official currency of Eurozone	Alphabets (3)
Amount	Order amount	Numeric (12, 2)

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

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

delivery_country	Shipping country	Alphabets (50) **Characters allowed:** Alphabet (A-Z), (a-z). Space in between words.
delivery_tel	Shipping phone number	Numeric (22)
merchant_param1	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param2	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param3	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param4	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)
merchant_param5	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the reconciliation report.	Alphanumeric (100) **Characters allowed:** Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen)

vault	This parameter can be used if merchant availing the vault option. On using vault functionality if card details are saved at KOTAK end value returned will be Y. If card details are not saved at KOTAK end the value returned for this parameter will be N	Character(1) **Characters allowed:** Y or N
offer_type	This parameter can be used for sending additional information if customer has used any discount or promotion while completing the transaction. If customer is using discount-coupon, value of this parameter would be discount. If customer is using promo-code, value of this parameter would be promotion.	Alphabets (9)
offer_code	This parameter can be used for sending additional information about the discount coupon and Promo code used while completing the transaction. If customer has used Discount the value sent would be Y or N accordingly. If customer has used Promotion the value sent would be Promo code	Alphanumeric (30)
discount_value	This parameter can be used for sending additional information about the discounted amount.	Numeric (12,2)
si_status	Status of the standing instruction request "0" denotes success. "1" denotes failure.	Numeric
si_sub_ref_no	This is reference number created by KOTAK for each new subscription on the KOTAK system. This is the number that must be sent with each new "on demand" charge to identify the customer.	Alphanumeric (15)
si_mer_ref_no	This is the unique identifier send by the merchant in the request. E.G. For insurance – Policy number. It can also be a customer reference number.	Alphanumeric (30)
si_error_desc	Reason for failure to setup SI.	Alphanumeric (150)

retry	This parameter can be used if merchant availing the retry option. If the transaction is processed through retry attempt returned value will be Y. If the transaction is not processed through retry attempt returned value will be n.	Character(1) **Characters allowed:** Y or N
response_code	This parameter contains the code for each bank response message.	Numeric

Contact Details

Kotak Technical Support
Email ID: service@ccavenue.com
Contact No: 02267425555 Extn: 418/419/420