



Home



Getting Started

## WEB INTEGRATION



Paytm Checkout



Plugins



Hosted E-commerce Websites

## MOBILE INTEGRATION



Android



iOS

## ENTERPRISE INTEGRATIONS



Auto Debit



Pre-Auth



API Reference



Delayed Notification



Refund Management



FAQs

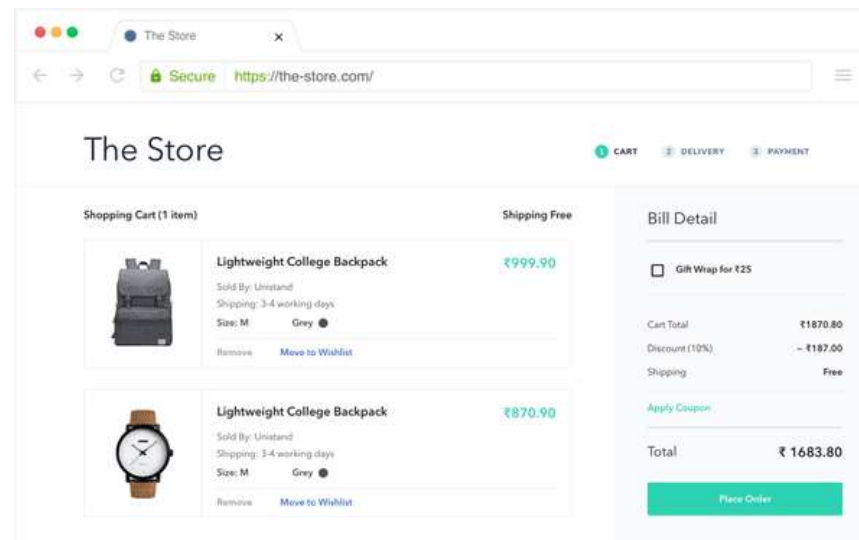


Testing Integration

## Collect online payments with Paytm Payment Gateway

Paytm Checkout for your website provides a secure, PCI-compliant way to accept Debit/Credit card, Net-Banking, UPI and Paytm wallet payments from your customers.

### Demo for Paytm checkout



### Overview of payment processing via Paytm checkout

The payment process starts at the click of pay button on merchant order summary page. On this click, you need to:

1. Create an order in your order system and then generate checksumhash at your server end for payment request. Checksumhash is used for detecting errors or tampering introduced during its transmission of request. Checksum is generated using merchant Key which should be kept only on server side for security reasons
2. Post the payload and checksumhash in an HTML form POST on Paytm's server. This redirects the customer to Paytm's payment page
3. Customer fills payment details and completes the payment authentication. Once the payment is complete, response is posted in HTML form POST on your website's callback URL
4. Verify checksumhash received in response to ensure that it has not been tampered
5. Lastly, verify transaction status with [Transaction Status API](#) via server to server call. This protects you from scenarios where your account credentials are

Home

Getting Started

## WEB INTEGRATION

Paytm Checkout

Plugins

Hosted E-commerce Websites

## MOBILE INTEGRATION

Android

iOS

## ENTERPRISE INTEGRATIONS

Auto Debit

Pre-Auth

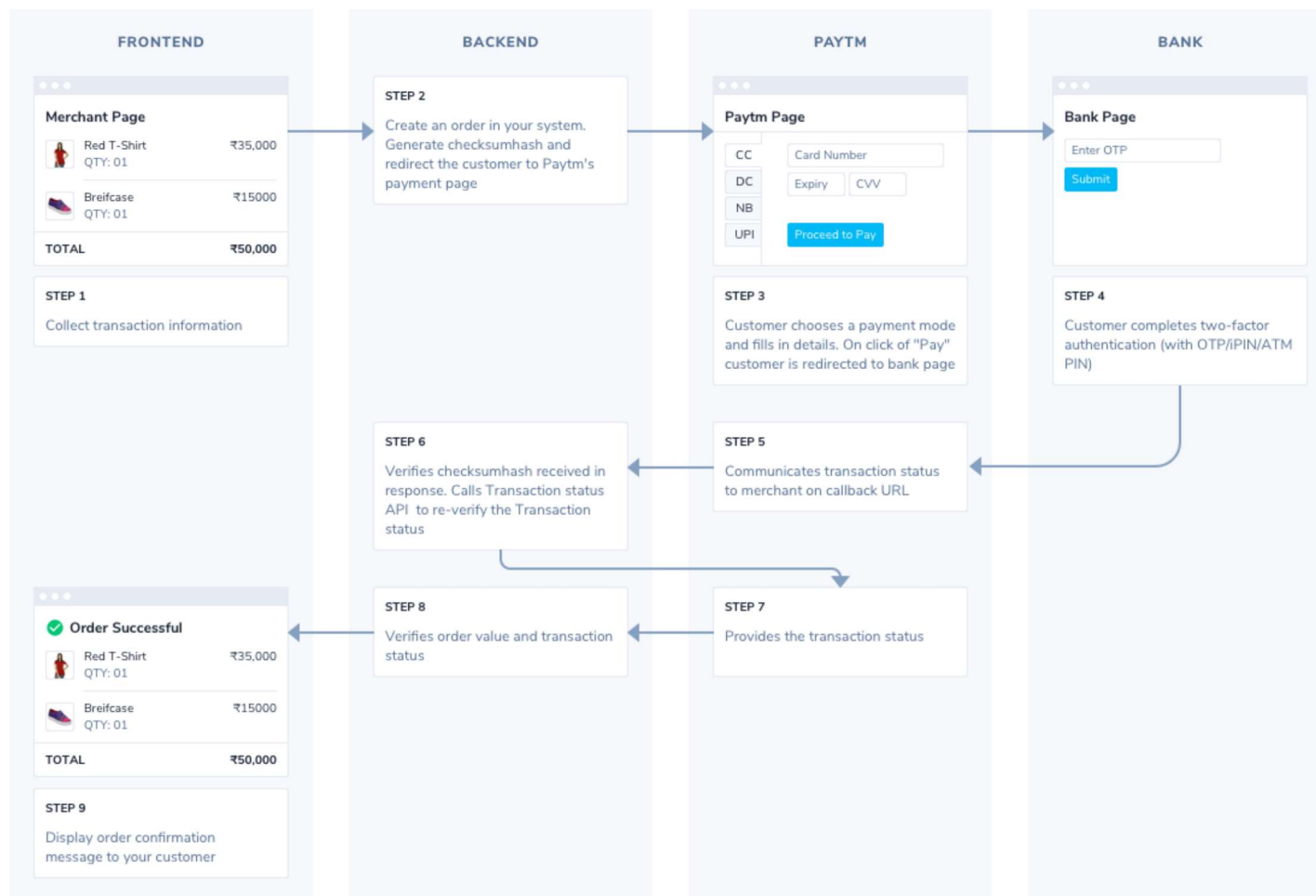
API Reference

Delayed Notification

Refund Management

FAQs

Testing Integration



## Steps in processing payments via Paytm checkout

## Step 1 :

At the click of payment button by customer on your website, create an order in your system and generate the required payload for payment request. Parameters of payload are provided below

## REQUEST ATTRIBUTES

MID

This is a unique identifier provided to every merchant by Paytm. MID is part of your account credentials and



Home



Getting Started

**WEB INTEGRATION**

Paytm Checkout



Plugins



Hosted E-commerce Websites

**MOBILE INTEGRATION**

Android



iOS

**ENTERPRISE INTEGRATIONS**

Auto Debit



Pre-Auth



API Reference



Delayed Notification



Refund Management



FAQs



Testing Integration

ORDER_ID String(50) Mandatory	Unique reference ID for a transaction which is generated by merchant Special characters allowed in Order ID are: @, -, _ , \$ , .
CUST_ID String(64) Mandatory	Unique reference ID for every customer which is generated by merchant Special characters allowed in Cust_ID are @, ! , _ , \$ , .
TXN_AMOUNT String(10) Mandatory	Amount in INR payable by customer. Should contain digits up to two decimal points. Only special character allowed is (".")
CHANNEL_ID String(3) Mandatory	This parameter is used to control the theme of the payment page. Based on the channel passed, Paytm will render the layout suitable for that specific platform For websites, the value is WEB For Mobile websites/App, the value is WAP
WEBSITE String(30) Mandatory	For staging environment: For staging environment: WEBSTAGING For production environment: Will be available <a href="#">here</a> once your activation is complete
CHECKSUMHASH String(108) Mandatory	Signature to avoid tampering. Generated using server side checksum utility available <a href="#">here</a>
MOBILE_NO String(15) Optional	Customer's mobile number. Passing this enables faster login for customer into his/her Paytm account. When the customer clicks on login, the mobile number comes pre-filled on our payment page
EMAIL String(50) Optional	Customer's email ID
INDUSTRY_TYPE_ID String(20) Mandatory	For staging environment: "Retail" For production environment: Will be available <a href="#">here</a> once your activation is complete
CALLBACK_URL String(255) Mandatory	On completion of transaction, Paytm payment gateway will send the response on this URL.. Sample URL is - <a href="https://merchant.com/callback/">https://merchant.com/callback/</a>
PAYMENT_MODE_ONLY String(2) Optional	If merchant wants to allow payment mode selection on his website or restrict the payment to a particular paymode, then value to be passed is "YES". For example, with this parameter merchant can ensure that customer only uses credit card as a paymode.



Home



Getting Started

**WEB INTEGRATION**

Paytm Checkout



Plugins



Hosted E-commerce Websites

**MOBILE INTEGRATION**

Android



iOS

**ENTERPRISE INTEGRATIONS**

Auto Debit



Pre-Auth



API Reference



Delayed Notification



Refund Management



FAQs



Testing Integration

AUTH\_MODE  
String(10) Conditional

Required If PAYMENT\_MODE\_ONLY = YES, then  
For Credit/Debit card - 3D  
For Wallet, Net Banking – USRPWD

PAYMENT\_TYPE\_ID  
String(15) Conditional

Required If PAYMENT\_MODE\_ONLY = Yes, then  
Credit card payment mode – CC  
Debit card payment mode - DC  
Net banking payment mode - NB  
Paytm wallet – PPI  
EMI - EMI  
UPI - UPI

BANK\_CODE  
String(5) Conditional

Required If PAYMENT\_MODE\_ONLY = Yes PAYMENT\_TYPE\_ID = NB  
List of Bank Codes provided in this [PDF](#)

**Step 2:**

Generate checksumhash using Paytm library with parameters in key value pairs. Using the payload and checksumhash make an HTML form post and redirect customer to Paytm server. Code snippets and Github links for the checksum utility and HTML form post are provided below.

**JAVA .NET PHP**

```
String merchantMid = "rxazcv89315285244163";
// Key in your staging and production MID available in your dashboard
String merchantKey = "gKpu7IKaLSbkchFS";
// Key in your staging and production merchant key available in your dashboard
String orderId = "order1";
String channelId = "WEB";
String custId = "cust123";
String mobileNo = "7777777777";
String email = "username@emailprovider.com";
String txnAmount = "100.12";
String website = "WEBSTAGING";
// This is the staging value. Production value is available in your dashboard
String industryTypeId = "Retail";
// This is the staging value. Production value is available in your dashboard
String callbackUrl = "https://<Merchant_Response_URL>";
TreeMap<String, String> paytmParams = new TreeMap<String, String>();
paytmParams.put("MID", merchantMid);
paytmParams.put("ORDER_ID", orderId);
```



Home



Getting Started

## WEB INTEGRATION



Paytm Checkout



Plugins



Hosted E-commerce Websites

## MOBILE INTEGRATION



Android



iOS

## ENTERPRISE INTEGRATIONS



Auto Debit



Pre-Auth



API Reference



Delayed Notification



Refund Management



FAQs



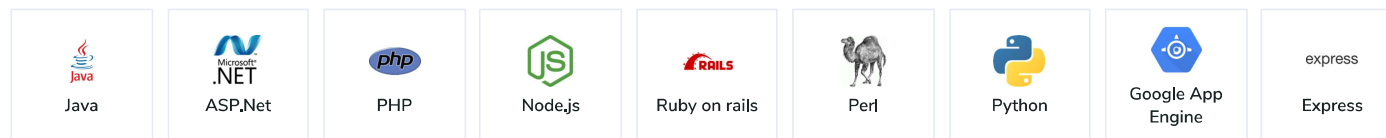
Testing Integration

## Endpoints:

Staging: <https://securegw-stage.paytm.in/theia/processTransaction>Production: <https://securegw.paytm.in/theia/processTransaction>

Get the sample code for a language of your choice -

## For Web:



## For App:



## HTML Form Post

```
<html>
  <head>
    <title>Merchant Check Out Page</title>
  </head>
  <body>
    <center><h1>Please do not refresh this page...</h1></center>
    <form method="post" action="https://securegw-stage.paytm.in/theia/processTransaction?ORDER_ID=order1" name="f1">
      <table border="1">
        <tbody>
          <input type="hidden" name="MID" value="rxazcv89315285244163">
          <input type="hidden" name="WEBSITE" value="WEBSTAGING">
          <input type="hidden" name="ORDER_ID" value="order1">
          <input type="hidden" name="CUST_ID" value="cust123">
          <input type="hidden" name="MOBILE_NO" value="7777777777">
          <input type="hidden" name="EMAIL" value="username@emailprovider.com">
          <input type="hidden" name="INDUSTRY_TYPE_ID" value="Retail">
          <input type="hidden" name="CHANNEL_ID" value="WEB">
          <input type="hidden" name="TXN_AMOUNT" value="100.12">
```



Home



Getting Started

## WEB INTEGRATION



Paytm Checkout



Plugins



Hosted E-commerce Websites

## MOBILE INTEGRATION



Android



iOS

## ENTERPRISE INTEGRATIONS



Auto Debit



Pre-Auth



API Reference



Delayed Notification



Refund Management



FAQs



Testing Integration

## Step 3:

Customer fills the payment details and is redirected to bank page for authorization. Once the transaction is authorized, Paytm receives the response from the bank and returns a status to your website via your callback URL. Response attributes description and sample HTML form post is provided below

RESPONSE  
ATTRIBUTES

MID String(20)	This is a unique identifier provided to every merchant by Paytm
TXNID String(64)	This is a unique Paytm transaction ID that is issued by Paytm for each transaction
ORDERID String(50)	Unique reference ID for a transaction which is generated by merchant and sent in the request
CUST_ID String(64)	Unique reference ID for every customer which is generated by merchant and sent in request
BANKTXNID String	The transaction ID sent by the bank. In case of Paytm proprietary instruments too, there is unique reference number generated by Paytm's system. In case the transaction does not reach the bank, this will be NULL or empty string. Primary reason for this is user dropping out of the payment flow before the transaction reaches to bank to servers
TXNAMOUNT String(10)	Amount paid by customer in INR
CURRENCY String(3)	Currency in which the transaction has taken place. Currently only "INR" is the supported currency of transaction
STATUS String(20)	This contains the transaction status and has only three values: TXN_SUCCESS, TXN_FAILURE and PENDING
RESPCODE String(10)	Codes refer to a particular reason of payment failure/success. List in this <a href="#">PDF</a>
RESPMSG	

Home

Getting Started

WEB INTEGRATION

Paytm Checkout

Plugins

Hosted E-commerce Websites

MOBILE INTEGRATION

Android

iOS

ENTERPRISE INTEGRATIONS

Auto Debit

Pre-Auth

API Reference

Delayed Notification

Refund Management

FAQs

Testing Integration

TXNDATE DateTime	Date and Time of transaction in the format "yyyy-MM-dd HH:mm:ss.S"Example: "2015-11- 02 11:40:46.0"
GATEWAYNAME String(15)	Gateway used by Paytm to process the transactions. By paymodes, the details are provided below Credit, debit cards UPI - Gateway used to process the transaction. For example, if HDFC gateway has been used to process SBI credit card transactions, the value will be HDFC Net banking - Netbanking transactions are not routed via gateway. Hence issuing bank name is passed in this field Paytm Wallet - Wallet
BANKNAME String(500)	Name of issuing bank of the payment instrument used by customer. By paymodes, the details are provided below Credit, debit cards, net banking - Name of the issuing bank. Example in case customer uses SBI's credit card, the value will be "SBI" Paytm Wallet - Wallet Note that in case of UPI - This parameter will be not be present in the response
PAYMENTMODE String(15)	The payment mode used by customer for transaction Credit card – CC Debit card - DC Net banking - NB UPI - UPI Paytm wallet – PPI
CHECKSUMHASH String(108)	Security parameter to avoid tampering. Verified using server side checksum utility provided by Paytm. Utilities to generate checksumhash is available <a href="#">here</a>
BIN_NUMBER String(6)	Starting 6 digit of Credit Card/Debit Card which used in transaction. For other modes, this field not provided in response
CARD_LAST_NUMS String(4)	Last 4 digit of Credit Card/Debit Card which used in transaction. For other modes, this field not provided in response
<pre>&lt;html&gt; &lt;head&gt;   &lt;meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"&gt;   &lt;title&gt;Paytm Secure Online Payment Gateway&lt;/title&gt; &lt;/head&gt;</pre>	



Home



Getting Started

## WEB INTEGRATION



Paytm Checkout



Plugins



Hosted E-commerce Websites

## MOBILE INTEGRATION



Android



iOS

## ENTERPRISE INTEGRATIONS



Auto Debit



Pre-Auth



API Reference



Delayed Notification



Refund Management



FAQs



Testing Integration

```

<tr>
<td><STRONG>Transaction is being processed,</STRONG></td>
</tr>
<tr>
<td><font color='blue'>Please wait ...</font></td>
</tr>
<tr>
<td>(Please do not press 'Refresh' or 'Back' button</td>
</tr>
</table>

```

## Step 4:

Checksumhash received in response of transaction needs to be verified on merchant server using Paytm library with all the parameters in key value pairs. Code snippets and Github links for the checksum utility are provided below

**JAVA**   **.NET**   **PHP**

```

private final String merchantKey = "gKpu7IKaLSbkchFS";
private String paytmChecksum = null;
// Create a tree map from the form post param
TreeMap<String, String> paytmParams = new TreeMap<String, String>();
// Request is HttpServletRequest
for (Entry<String, String[]> requestParamsEntry : request.getParameterMap().entrySet()) {
    if ("CHECKSUMHASH".equalsIgnoreCase(requestParamsEntry.getKey())){
        paytmChecksum = requestParamsEntry.getValue()[0];
    } else {
        paytmParams.put(requestParamsEntry.getKey(), requestParamsEntry.getValue()[0]);
    }
}
// Call the method for verification
boolean isValidChecksum = CheckSumServiceHelper.getCheckSumServiceHelper().verifycheckSum(merchantKey, paytmParams, payt
// If isValidChecksum is false, then checksum is not valid
if(isValidChecksum){
    System.out.append("Checksum Matched");
}else{

```

Get the sample code for a language of your choice -

**For Web:**





Home



Getting Started

## WEB INTEGRATION



Paytm Checkout



Plugins



Hosted E-commerce Websites

## MOBILE INTEGRATION



Android



iOS

## ENTERPRISE INTEGRATIONS



Auto Debit



Pre-Auth



API Reference



Delayed Notification



Refund Management



FAQs



Testing Integration



Perl



PHP



Node.js



Java



ASP.Net



Python



Ruby on rails

Google App  
Engine

Express

## For App:



PHP



Node.js



Java



ASP.Net



Python



Ruby on rails

Google App  
Engine

## Step 5 :

Validate transaction response via server side request using [Transaction Status API](#). This API requires checksumhash in request and its verification in response. The status should be treated as the final status of the transaction

## On completion of your integration -

Post completion of integration on your staging environment, do a complete transaction from order summary page on your website or mobile app

- 1 Attempt a test transaction using [test paymodes credentials](#)
- 2 Ensure you re-verify transaction response with [Transaction Status API](#) via server to server call in payment flow and not separately as a one time activity
- 3 See the transaction details in "Test Data" mode on your [dashboard](#)

Once the test transaction is complete, move your code to live environment with production account details. Note that production accounts details are available after you have [activated your account](#) on the dashboard

Lastly, it's recommended that you read about [Managing Refunds](#) and [late payment notifications](#)

In case of any issues with integration, please [get in touch](#)