freecharge

Merchant Integration Guide

Web and Mobile

FreeCharge Payment Technologies Private Limited
Registered office: Ground Floor, Plot 237, Okhla Industrial Estate, New Delhi 110020
Email ID – merchantsupport@freecharge.com

Document Name: Merchant Integration Guide – Web and Mobile
Document Version: V2.11
Version History:

Version	Date	Action
1.0	24/11/2015	Created
1.1	03/12/2015	API set has been changed compared to previous version
2.0	21/11/2015	APIs added to the existing list
2.1	31/12/2015	Checksum process added to the list
2.2	08/02/2016	Checksum process modified
2.3	08/02/2016	Checksum example modified
2.4	25/02/2016	Sandbox information modified
2.5	02/03/2016	Process Payment API variable updated/added/modified
2.6	07/03/2016	Checksum Utility links added
2.7	09/03/2016	API - Get Transaction By ID API (modified)
2.8	18/03/2016	Status for Process Payment API modified
2.9	12/04/2016	Added remember me cookie code (Android specific)
2.10	13/04/2016	New Error codes added
2.11	20/04/2016	Modified FAQ s and Checksum

Document Author: Vandana Pandey
Document Approver: Amit Kurseja
SPOC Information: merchantsupport@freecharge.com

Glossary

Abbreviation	Description
API	Application Program Interface
FC	FreeCharge
MID	Merchant Identity

Audience

This document is intended to be used by the merchant developers to integrate FreeCharge Payment product with the merchant checkout flow and enable transactions for customers.

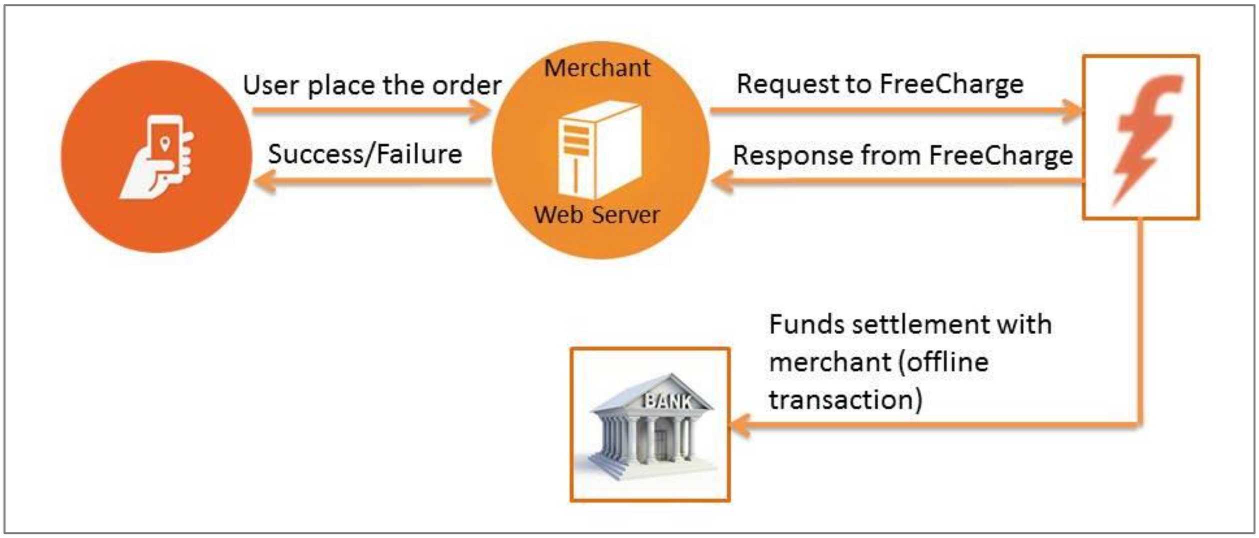
Purpose of document

This document helps you:

- To integrate FreeCharge Payments product in your checkout flow
- To implement and test transactions in sandbox environment
- To understand different types of error codes you might encounter during integrating or in production
- To provide a set of Frequently Asked Questions and support contact points for easy implementation

Process Flow

The following diagram broadly explains the process flow for transactions through Freecharge.



Sandbox and Production Environment

Sandbox environment enables you to completely test your payment integration before you go live with the FreeCharge Payment option. You can thoroughly test all the payment scenarios and refunds in the Sandbox.

When you test your implementation in the Sandbox mode you will be able to:

- Simulate the buyer experience for payment
- Simulate the error conditions to manage your buyers' experiences during the online transaction experience,
- Simulate a payment refund
- Simulate transaction history and status tracking
- Simulate API operation calls to ensure that the calls are configured correctly and the responses include all the payment parameters that are required to track the entire order

Note: User will be able to login with the FreeCharge account credentials and see their stored cards and FreeCharge balance. However, following will not be allowed:

- Actual wallet transaction
    - Wallet balance will be shown, wallet will not be debited for the transactions taking place
- Credit card and netbanking transactions will only be simulated and are not processed in actual

You will be able to see all the transactions that have been performed in sandbox and can simulate refunds on the same

URL – https://checkout-sandbox.freecharge.in

Production environment enables you to start making live transactions and receive money to your actual bank accounts. Your Production merchantId will be shared with your separately by an email.

URL – https://checkout.freecharge.in

Remember Me Cookie (applicable for Android)

To enable remember me cookie in order to make the user auto login, add the following code:

1. Go to https://bitbucket.org/freecharge/paywithfreecharge-android
2. Go to Classname **COLibWebViewClient**.
3. Add the following code:

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP)
            {
             android.webkit.CookieManager.getInstance().flush();
            }
           Else
```

```
{
 CookieSyncManager.getInstance().sync();
}
```

Integration Application Program Interfaces (APIs)

By integrating our APIs with your buy flow on your platform, you can enable payments for consumers. The user will be redirected to FreeCharge page to complete the payment. After payment completion, user will be redirected to URL specified by you.

APIs are a set of program used as an interface by software components to communicate with each other. An API may include specifications for routines, data structures, object classes, and variables.

The following APIs are available to integrate FreeCharge on merchant website/app:

- Process Payment
- Get Transaction by ID
- Refund

Authentication

To authenticate access to FreeCharge APIs, API request checksum is included in the request body. API request checksum must be passed as the authToken.

Generic Response structure

On success, you get an HTTP response code of 200 or 201, plus the requested JSON object in the body. Otherwise, you will get an HTTP status code (202) plus the JSON object below describing the error:

```
{
 response code:           "string",
 response message: "string"
}
```

HTTP status codes

The following are the HTTP status codes. The HTTP status codes are same for all the refunds and payments.

Error Code	Error Description	Method
200	OK	Post
202	Failure Message	Post

500	Internal Server Error	Internal server exception

Note: All the API's can send 202 in their http header response and it should be treated as an exception.

Field	**Description**	**Constraint**	**Mandatory(M)** **Optional (O)**
errorCode	Error code		M
errorMessage	Error Message		M

Checksum

The following are the steps to create checksum:

1. Create **JSON** of your request.
2. **JSON** must be alphabetically serialized and must not contain null or empty values.
3. To create a single String, **concat JSON** and **key** (merchant secret key).
4. To create checksum of concatenated String, use **SHA-256** algorithm.

Note: In order to create checksum for init/refunds/transactions, the request MUST contain ALL the fields which merchant is sending for that API (this should also have all the mandatory fields as mentioned in each API details of action (init, refund or transaction)

Example

The following is the example to create checksum for **payment init**:

jsonString

```
{ "merchantTxnId": "iTEST1_***sdsds%1233", "amount": "200.65", "currency": "INR", "furl":
"http://domain:8080//failure.html?ji=ered&sdee=3423", "surl":
"http://domain:8080//success.html?hi=ho&abc=abr", "productInfo": "auth", "email":
"test2323@test.com", "mobile": "9090909090", "customerName": "testCustomer", "customNote":
"please make it fast...; !", "channel": "WEB", "os": "ubuntu-14.04", "merchantId": "abcXXX123"}
```

Merchantkey

```
qee244-XXX-XXX-XXX-XXXX7897
```

Concatenate **json String with Key (Merchant secret key) as shown above**

```
{ "merchantTxnId": "iTEST1_***sdsds%1233", "amount": "200.65", "currency": "INR", "furl":
"http://domain:8080//failure.html?ji=ered&sdee=3423", "surl":
"http://domain:8080//success.html?hi=ho&abc=abr", "productInfo": "auth", "email":
"test2323@test.com", "mobile": "9090909090", "customerName": "testCustomer", "customNote":
"please make it fast...; !", "channel": "WEB", "os": "ubuntu-14.04", "merchantId":
"abcXXX123"}qee244-XXX-XXX-XXX-XXXX7897"}
```

Logic to Create Checksum

```
ava:
MessageDigest md;
 String plainText = jsonString.concat(Merchantkey);

 try        {
            md = MessageDigest.getInstance("SHA-256");
            } catch (NoSuchAlgorithmException e)
            {

            }
            md.update(plainText.getBytes(Charset.defaultCharset()));

            byte[] mdbytes = md.digest();

            // convert the byte to hex format method 1
            StringBuffer checksum = new StringBuffer();
            for (int i = 0; i < mdbytes.length; i++)
            {
            checksum.append(Integer.toString((mdbytes[i] & 0xff) + 0x100, 16).substring(1));
            }
            return checksum;
```

Final request with checksum

The following is the final request with checksum:

```
{ "merchantTxnId": "iTEST1_***sdsds%1233", "amount": "200.65", "currency": "INR", "furl": "http://domain:8080//failure.html?ji=ered&sdee=3423", "surl": "http://domain:8080//success.html?hi=ho&abc=abr", "productInfo": "auth", "email": "test2323@test.com", "mobile": "9090909090", "customerName": "tes tCustomer", "customNote": "please make it fast...; !", "channel": "WEB", "os": "ubuntu-14.04", "merchantId": "abcXXX123","checksum":"5fc487b4f95ec1a6ff9b0cdc939aa0112cb0b78b56fbac77ac23943d150a7c6f"}
```

Response Checksum Verification Process

1. You need to send the request along with the *Checksum* generated to FreeCharge Server.
2. FreeCharge server will process request and send you the response along with the *Checksum* generated on the basis of parameters mentioned in response.
3. You need to generate a *Checksum* based on the response sent by FreeCharge (Parameters mentioned in response excluding checksum, same process to be followed as creating checksum in request).
4. Match it with the *Checksum* sent by FreeCharge in response.

Note: In case of failure transactions, the value of checksum might be **null**.

Developer Kit & Server Side Utilities

The following are the utility links for checksum generation in different platforms like IOS, JAVA, PHP and others. You can download and used it according to the platform requirement without writing logic or code. All the validation has been added in these utility that will reduce error mistakes.

Platform	URL
IOS	https://bitbucket.org/freecharge/paywithfreecharge-ios
ANDROID	https://bitbucket.org/freecharge/paywithfreecharge-android
JAVA	https://bitbucket.org/freecharge/paywithfreecharge-java
PHP	https://bitbucket.org/freecharge/paywithfreecharge-php
PYTHON	https://bitbucket.org/freecharge/paywithfreecharge-python

Process Payment

Consumer will post all these parameter on given URL provided by FreeCharge.

Note: The Generic response headers and HTTP status codes will not be applicable for Process payment API.
All the parameters must be written in camelcase.

Post

api/v1/co/pay/init

Request

Parameter Name	Description	Type	Mandatory (M) Optional (O)
merchantId	Unique Id assigned by FreeCharge to merchant	String (127)	M
checksum	Generated using the utility provided		M
merchantTxnId	Unique Order Id generated by the merchant		M
amount	Order Amount to two decimal places e.g. “255.00”	String	M
channel	Channel through which request is being made	WEB WAP ANDROID IOS WINDOWS	M
surl	URL to redirect to in case of success	String	M
currency	Default = INR	String (3)	O
furl	URL to redirect to in case of failure	String	M
productInfo	Details of products comprising the order		O

email	Email of transacting customer		O
mobile	Mobile of transacting customer		O
customerName	Name of the customer	String	O
blockPaymentType	The payment types(CC/DC/NB) that user cannot use for making the payment	CC/DC/NB	O
metadata	Custom data that will be passed back to you with response	map of user defined field	O
pg	The payment options section that is displayed to the customer by default on the FreeCharge page. Hence, if PG='NB', then after redirection to FreeCharge Payment page, by default the Net Banking option will be displayed. (PG parameter may take different values like : NB for Net Banking tab, CC for Credit Card tab, DC for Debit Card tab, CASH for Cash Card tab and EMI for EMI tab) Note: PG = CC, that is Credit Card tab is recommended. If PG is left empty, CC is by default.		O

dropCategory	This parameter is used to drop a particular payment method. **Note**: Please make sure that you use this parameter only after proper testing. Else, It may lead to unwanted display of payment method	Format: CC|AMEX, DC|VISA, EMI|EMI6, DC|VISA|MAST, NB|ICIB	O
customNote	To display message to the customer on payment page.	Characters allowed: A to Z, a to z, 0 to 9, % (percentage), , (comma), . (decimal), ' (apostrophe)	O
noteCategory	It defines the selected payment category for customNote. Value for this field should be comma separated.	.	O
os			O
version			O

Description of the status

Status	Decription
Failed	Transaction has been completed and got failed due to some reason. This should be treated as Failed transaction. Example: incorrect card number or technical issues.
Completed	Transaction has been completed and has been succeeded.

Sample

```
<!doctype html>
```

```
<html>
<head>

    <title>Checkout</title>
   </head>
<body>
<form enctype='application/json' action="https://checkout-sandbox.freecharge.in/api/v1/co/pay/init
" id="searchForm"
method="post">
    <h1>Client Order Form</h1>
    <fieldset>
              <legend>Merchant / USER details</legend>
              <div>
              <label>merchantId<input id="merchantId" value=" " name="merchantId" type="text"
required></label>
              </div>
              <div>
              <label>merchantTxnId <input id="merchantTxnId" value="" name="merchantTxnId"
type="text" required></label>
              </div>
              <div>
              <label>amount <input id="amount" value="1000" name="amount" type="number"
required></label>
              </div>
              <div>
              <label>currency <input id="currency" name="currency" type="text" value=""
required></label>
              </div>
              <div>
              <label>furl <input id="furl" name="furl" type="text" value=" "  required></label>
              </div>
              <div>
              <label>surl <input id="surl" name="surl" type="text" value=" "  required></label>
              </div>
              <div>
              <label>productInfo <input id="productInfo" name="productInfo" type="text" value="auth"
></label>
              </div>
              <!--  this checksum field is to be calculated by the steps mentioned -->
                  <label>checksum <input id="checksum" name="checksum" type="text"    value=""
></label>
```

```
<div>
        <label>email <input id="email" value="example@domain.com" name="email" type="email" required></label>
    </div>
        <div>
        <label>mobile <input id="mobile" name="mobile" type="text" value=""  required></label>
        </div>
        <div>
        <label>customerName <input id="customerName" value=" " name="customerName" type="text" ></label>
        </div>
        <div>
        <label>blockPaymentType <input id="blockPaymentType" name="blockPaymentType" type="text" value="" ></label>
        </div>
        <div>
        <label>pg <input id="pg" name="pg" type="text" value=""  ></label>
        </div>
        <div>
        <label>dropCategory <input id="dropCategory" name="dropCategory" type="text" value="" ></label>
        </div>
        <div>
        <label>customNote <input id="customNote" name="customNote" type="text" value="" ></label>
        </div>
        <div>
        <label>noteCategory <input id="noteCategory" name="noteCategory" type="text" value="" ></label>
        </div>
        <div>
        <label>channel <input id="channel" name="channel" type="text" value=""  ></label>
        </div>
        <div>
        <label>metadata <input id="metadata" name="metadata" type="text" value=""  ></label>
        </div>
  </fieldset>
  <button id="checkout" class="btn btn-lg btn-block signin">Checkout</button>
  <p id="msg"></p>
</form>
</body>
```

```
</html>
```

Response

Response will be posted to merchant on success URL given by the consumer.

Parameter Name	Description	Type	Mandatory (M) Optional (O)
Status	Status of transaction	FAILED,COMPLETED	M
txnId	FreeCharge Transaction Id (may be null in case of failure)	String	O
merchantTxnId	Unique Order Id generated by the merchant	String	M
Mode	It describes the mode by which transaction was done, for example, CC, DC		
Amount	Amount Processed to two decimal places e.g. “255.00”	String	M
pgUsed	It contains the parameter of payment gateway used for transactions		O
pgUnderlier	Gateway(switch) used by the payment gateway aggregator for processing the transaction		O
checksum	Generated by FreeCharge and to be validated at merchant for validating	String	M
errorCode	This field is valid in case of failed transaction		M

errorMessage	This field is valid in case of failed transaction		M
currency	Default = INR	String (3)	O
email	Email of transacting customer	Either email or mobile is mandatory	O
mobile	Mobile of transacting customer	Either email or mobile is mandatory	O
metadata	Merchant defined data, same as passed in the request		O

Get Transaction by ID

This API helps you to get the updated transaction status for specific transactions Id.

Status	Description
Initiated	As soon as we get request for refund. Transaction status moves to Initiated state. In this case, merchant should retry to get status after 30 minutes.
Failed	Failure of refund takes place due to multiple reasons. Example: refunded amount is greater than the actual amount, technical issues.
Success	Refund Successful.
Pending	Transaction is being preceded. In this case, merchant should retry for getting status after 30 minutes. Example: some amount got deducted either through wallet or through payment gateway and transaction got failed. In this case, refund should be initiated for the user account. It’s in pending state.

Get

/api/v1/co/transaction/status?merchantId=&txnId=&checksum=&txnType=

Request

Parameter Name	Description	Type	Mandatory (M) Optional (O)
checksum	Required for authentication, information will be shared in a separate document		M
merchantId	ID generated for the merchant in actual	String (127)	M
txnType	CUSTOMER_PAYMENT or CANCELLATION_REFUND	String	M
merchantTxnId	Either merchantTxnId or txnId is mandatory	String (127)	O
txnId	Either merchantTxnId or txnId is mandatory	String (127)	O

Response

Parameter Name	Description	Type	Mandatory (M) Optional (O)
txnId	FreeCharge Transaction Id	String (127)	M
merchantTxnId	Unique Order Id generated by merchant	String	M
checksum	Checksum generated by FreeCharge for response object	String	M
amount	Amount processed	String	M
status	Transaction status	*INITIATED, FAILED SUCCESS, PENDING*	M

Refund Transaction

This API helps you to request a refund for a particular transaction. The following are the different status:

Description of the status

Status	Description
Initiated	As soon as we get request for refund. Transaction status moves to Initiated state. In this case, merchant should retry to get status after 30 minutes.
Failed	Failure of refund takes place due to multiple reasons. Example: refunded amount is greater than the actual amount, technical issues.
Success	Refund Successful.

Post

/api/v1/co/refund

Request

Parameter Name	Description	Type	Mandatory (M) Optional (O)
checksum	Required for authentication, information will be shared in a separate document		M
merchantId	ID generated for the merchant in actual	String (127)	M
txnId	FreeCharge Transaction Id	Either txnId or merchantTxnId must be specified	O
refundMerchantTxnId	Unique Order Id generated by Merchant for refund	String	M
merchantTxnId	Unique Order Id generated by Merchant	Either txnId or merchantTxnId must be specified	O

refundAmount	Amount to be refunded to two decimal places e.g. “255.00”	Must be less than the transaction amount	M
comments	Description for the refund		O

Sample

```
{
"refundAmount":null,
“refundMerchantTxnId”:null,
"comments":null,
"checksum":null,
"merchantId":null,
"merchantTxnId":null,
"txnId":null
}
```

Response

Parameter Name	Description	Type	Mandatory (M) Optional (O)
merchantTxnId	Merchant Transaction Id for the original transaction	String (127)	M
RefundTxnId	FreeCharge Transaction Id for the refund transaction	String (127)	M
refundedAmount	Amount refunded	Float	M
Checksum	Checksum generated by FreeCharge for response object		
RefundMerchantTxnId	Merchant Transaction Id for refund		
Status	Refund Status	INITIATED,SUCCESS FAILED	M

Sample

```
{
"amount":null,
"globalTxnId":null,
"merchantTxnId":null,
"status":null,
"success":false,
"checksum:null"
}
```

Error Codes

The following are the error codes:

Error Code	Error Description
E000	No error in processing request
E001	Invalid merchant Id
E002	Merchant not allowed for transaction
E003	Invalid merchant transaction id
E004	Duplicate merchant transaction id
E005	Invalid checksum/checksum mismatch
E006	Invalid request IP
E007	Request data contains invalid characters
E008	No transaction found for given txnId or merchantTxnId
E018	Application error occurred
E023	Invalid channel
E024	Amount precision should be max 2 digit
E026	Either txnId OR merchantTxnId should be present in request
E027	refundMerchantTxnId should not be blank

E100	Error processing request
E102	Invalid payment type code
E103	Invalid PG
E104	Amount not parsable
E105	Negative amount
E108	User entered wrong invalid card number
E109	User entered wrong CVV number
E113	Empty/Invalid Bank Code
E114	Invalid Bank Code for specified PG
E121	Merchant transaction id longer than allowed length
E300	Invalid refund amount
E301	Refund processing failure
E600	Empty Merchant Identity
E601	Empty Merchant Transaction Identity
E603	Empty Failure URL
E604	Empty Success URL
E615	Issue occurred while processing. Please contact customer support.
E616	OAuth not enabled for merchant
E617	Auth code is blank
E618	access token is blank
E619	Invalid auth code
E620	Invalid access token
E621	Invalid grant type
E622	Refresh token is blank

E623	Grant type is blank
E624	Refresh token is invalid
E625	Amount cannot be greater than wallet balance
E626	Customer payment is not in success state, refund is not possible
E627	Same idempotency Id For Different Refund Request
E628	Transaction amount must be greater than 1
E629	Refund Amount Exceeding Transaction Amount

Frequently Asked Questions

The following are the frequently asked questions related to the integration process:

1. **How can I go Live?**
   For live websites, integration should be a quick process. Simply integrate the FreeCharge APIs on your platform or connect your developer(s) with FreeCharge, if required.

2. **How do I integrate?**
   This document gives a detailed explanation of integration process. In case you face any confusion during any step, reach out to us at merchantsupport@freecharge.com

3. **How do I see transaction details?**
   You can view the transaction details and complete transaction history by using the Get All Transactions API.

4. **How do I refund transactions?**
   Refund can be initiated for the completed transactions, where user wants to get their money back due to pre-defined reasons.
   To initiate the refund process, see Refund Transaction API as mentioned above in the document.

5. **What is merchantId? How do I get it?**
   The merchantId is a unique identifier generated by FreeCharge and shared with you during the onboarding process.

6. **How do I generate Checksum?**
   Checksum information will be shared separately. If you do not receive it, write an email to merchantsupport@freecharge.com

7. **Do we get test cards to perform testing in sandbox environment?**

We do not provide any test cards for testing purpose. However, in sandbox environment you can get simulated responses for all the requests that are send from your server to FreeCharge server. Also, in production environment you can test with your own cards and then can refund the amount as required.

8. **Do we get any panel to access see the transactions in production or sandbox environment.**
   You can login to merchant panel (after onboarding is complete) and check the transactions that have been performed in the production environment. We do not have any panel for sandbox environment.

9. **Do we have any panel to initiate refund of a single transaction or in bulk.**
   You can login to merchant panel (after onboarding is complete) and initiate refund for a single transaction or many transactions at one go.

10. **How do I verify Checksum that is sent by FreeCharge Server in response?**
    You need to follow the following steps to ensure verification of Checksum:
    1. You need to send the request along with the *Checksum* generated to FreeCharge Server.
    2. FreeCharge server will process request and send you the response along with the *Checksum* generated on the basis of parameters mentioned in response.
    3. You need to generate a *Checksum* based on the response sent by FreeCharge (Parameters mentioned in response excluding checksum, same process to be followed as creating checksum in request).
    4. Match it with the *Checksum* sent by FreeCharge in response.
       *Note: In case of failure transactions, the value of checksum might be **null**.*

11. **I have tested the integration in sandbox environment, how do I initiate transaction in production environment.**
    After you have completed testing on Sandbox, a new production MID and secret will be shared with you by email. You can access the production environment at https://checkout.freecharge.in and start accepting live transactions.

12. **How do I get support?**
    Simply write to us at merchantsupport@freecharge.com and one of our engineers will connect with you.