NEPAL
CLEARING
HOUSE

connectIPS
Process and Interface Document
For Merchants/Technical Member

Version 2.0

Sept 2018

Table of Contents

Document Control

Document Name:	connectIPS Process and Interface Document For Merchants/Technical Member
Document Number:	**OP/PL/042**
Document Owner:	System Manager

Document Version	Date	Created/ Modified By	Reviewed By	Approved By	Remarks/ Amendment
Ver 1.0	September 2017	System Manager	Operation Manager	CEO	Draft Created
V 2.0	September 2018	System Manager	Senior System Analyst Operation Manager	CEO	Addition of Payment Validation and Transaction Reporting API details.
V 3.0	October 2018	System Manager	Senior System Analyst Operation Manager	CEO	Addition of revenue payment APIs

1. Introduction

1.1. Purpose

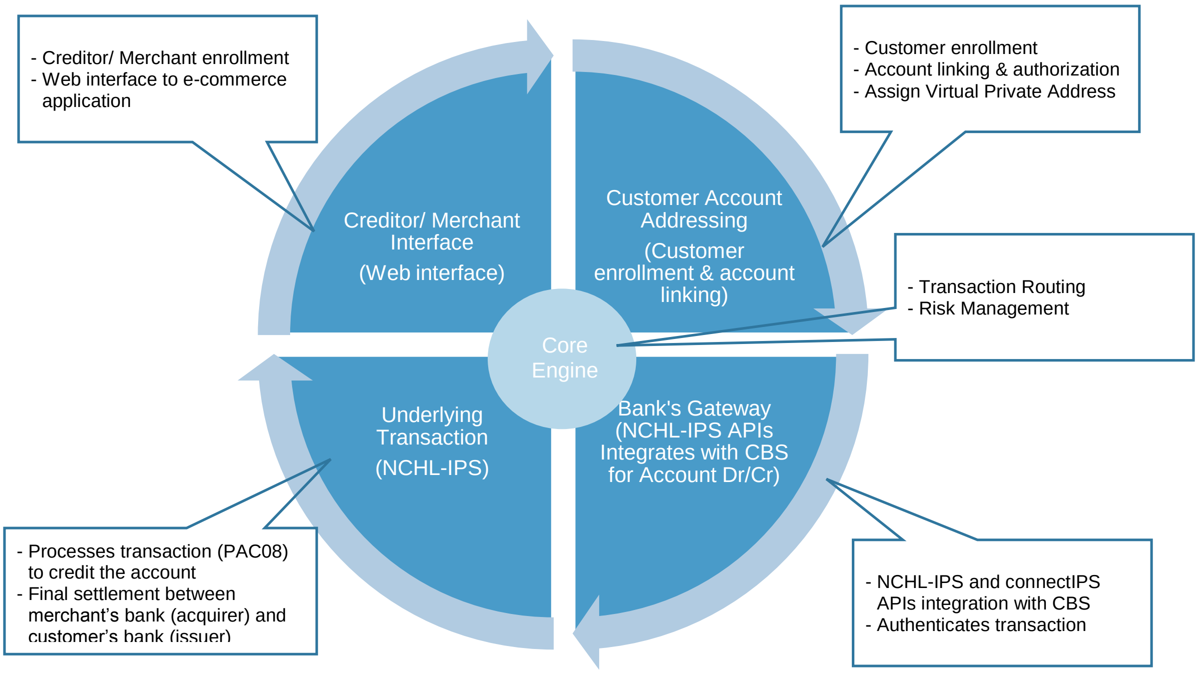
The purpose of the document is to describe necessary information for integrating merchant’s e-commerce site to *connectIPS* Core Module to initiate e-payment transactions by its customer.

1.2. Intended Audience and Reading Suggestions

The intended audience for this document comprises mainly software developers, software development managers or any other person with technical expertise in software development who is a Merchants’ technical partner or employee or has the Merchants’ approval for reading this document. And this process details are intended for the operations/ business managers of the merchants having e-commerce application.

2. Main Components of connectIPS

The *connectIPS* will have following functional components

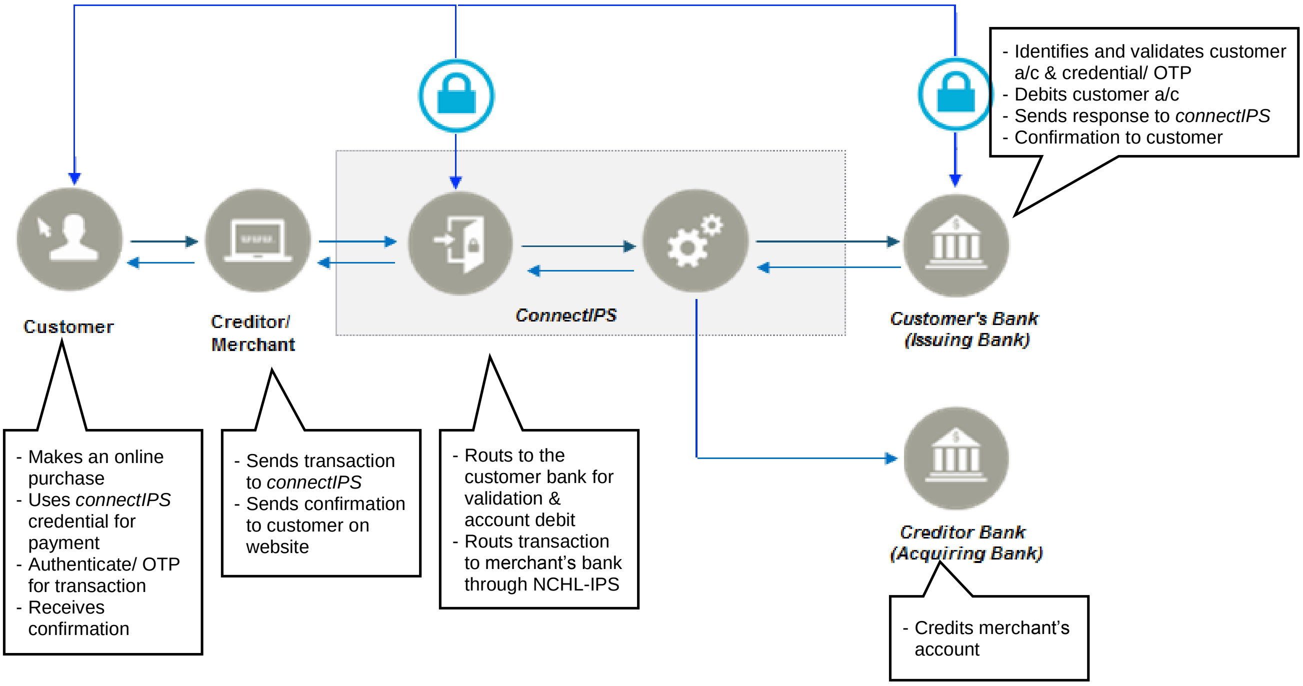

- Creditor/ Merchant enrollment
- Web interface to e-commerce application
- Customer enrollment
- Account linking & authorization
- Assign Virtual Private Address
Creditor/ Merchant Interface
(Web interface)
Customer Account Addressing
(Customer enrollment & account linking)
- Transaction Routing
- Risk Management
Core Engine
Underlying Transaction
(NCHL-IPS)
Bank's Gateway (NCHL-IPS APIs Integrates with CBS for Account Dr/Cr)
- Processes transaction (PAC08) to credit the account
- Final settlement between merchant's bank (acquirer) and customer's bank (issuer)
- NCHL-IPS and connectIPS APIs integration with CBS
- Authenticates transaction

The scope of this document for the merchants is Creditor Merchant interface, however information with regards to the overall transaction workflow has also been explained in the following sections.

3. *connectIPS* Core Module

The Core Module allows the banks to enroll their customers and their merchants and then such customers to initiate and process their online payment transactions. Some of the typical business cases that will be supported by *connectIPS* core module are C2G (eg, tax payment), C2B (utility payments), P2P, etc.

3.1. *connectIPS* Core Module Framework


Customer
Creditor/ Merchant
ConnectIPS
Customer's Bank (Issuing Bank)
Creditor Bank (Acquiring Bank)
- Identifies and validates customer a/c & credential/ OTP
- Debits customer a/c
- Sends response to *connectIPS*
- Confirmation to customer
- Makes an online purchase
- Uses *connectIPS* credential for payment
- Authenticate/ OTP for transaction
- Receives confirmation
- Sends transaction to *connectIPS*
- Sends confirmation to customer on website
- Routs to the customer bank for validation & account debit
- Routs transaction to merchant's bank through NCHL-IPS
- Credits merchant's account

3.2. Process/ Workflow of *connectIPS* Core Module

3.2.1. Merchant Enrollment

1. Interested merchant having e-commerce application can enroll their application for *connectIPS* through its bank by providing necessary supporting documents/ information. The bank will do the necessary KYC of the merchant and then send the details for creditor listing to NCHL for technical enrollment.
2. The creditor/ merchant will be activated from the Merchant Enrollment page and the APIs exposed to the merchant for necessary integration in their e-commerce application.
3. The system will send an email to the designated email id with Merchant Id and Password along with other information required to carry out its transaction.
4. The merchant will be able to login into the web interface and manage its operation from that portal for the needed reporting/ reconciliation.
5. It will support multiple accounts for multiple applications of a single merchant/ creditor. i.e. each e-commerce application will be linked to single bank account only, but a merchant can establish multiple such applications.

3.2.2. Customer Enrollment

1. Customer will access https://connectips.com portal for initial enrollment for the service. Alternatively, the customer bank can do the initial enrollment for the customer.

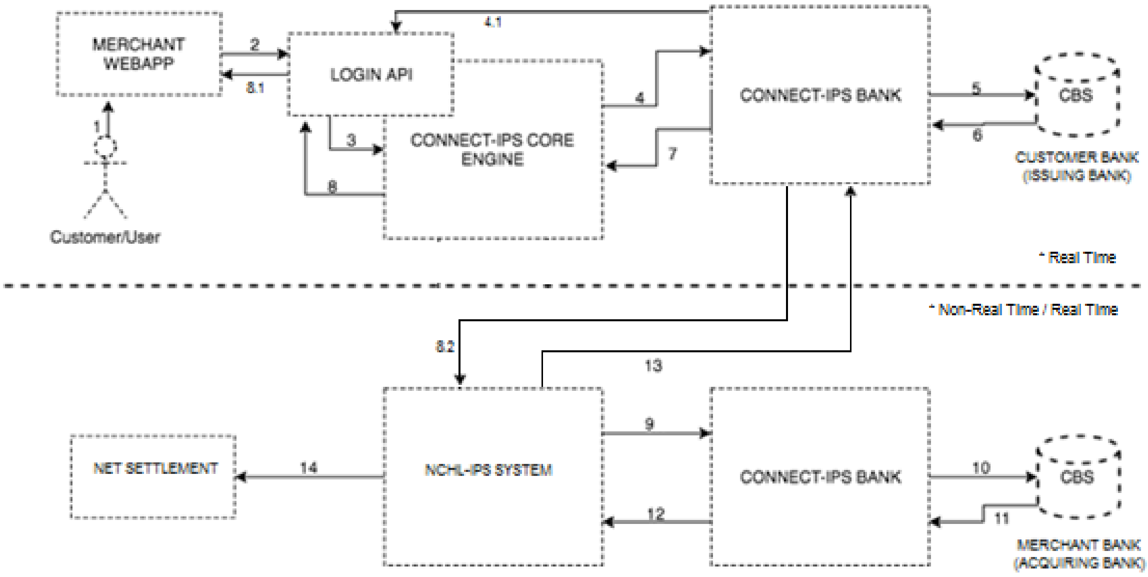
2. Customer will enter minimum information - Name, User Id (Mobile No), New Password and Email Id for enrollment. Mobile number will be the same as that provided to the banks (account number that will be linked later).
3. Upon successful enrollment, the customer will receive a confirmation in his/her Mobile No and/or Email Id.
4. This login id/ password will be used later for initiating payments through *connectIPS*, but will not be activated unless bank account is linked and activated by the respective bank.

3.2.3. Link Bank Account

1. Customers enrolled for *connectIPS* will be allowed to initiate transaction only after its bank account is linked and approved by the respective bank. However, each customer can link multiple bank accounts, each of which is uniquely identified by an "Account Identifier".
2. Account Identifier will be in the format of <mobile number>@<bank short code>@<Incremental value> and system generated.
3. The activation of the customer bank account will be done by the respective bank. It will be the responsibility of the bank to verify necessary customer account information/ status.
4. All such active customer account details against each User Id will be stored in the "Central Account Mapper" in the *connectIPS* core engine. Account Identifier will be used later for routing of transactions.
5. If Account Identifier is not specified in the transaction and in case of multiple bank accounts being linked, the transaction will be routed to the primary account selected by the user.

3.2.4. Transaction Flow

The Transaction flow of the *connectIPS* is depicted in the following diagram



1. The Customer will select "Pay By *connectIPS*" while checking out from merchants web application.
2. The customer will enter the *connectIPS* login credentials.
3. The customer will have to select the transaction account (in case of multiple accounts) and press submit after checking all the transaction information.
4. The *connectIPS* core engine will route the transaction to the respective bank for authorization.
   4.1. *connectIPS*-Bank has to open API for authorization (which is already available) which will retrieve the registered mobile number from the customer account and sends OTP to the mobile number using banks' SMS Gateway. In case bank do not have the provision to send OTP from their own SMS Gateway, the same can be pre-configured for sending from *connectIPS* core engine.

- *4.2.* Once customer receives OTP, he/she will enter the OTP in the customer portal for authentication and transaction confirmation.

5. *connectIPS* then redirects the request to the *connectIPS*-Bank, which will first validate the OTP and sends the debit transaction to the core banking system (CBS) using ISO8583 message.
6. The CBS will send the transaction response to the connectIPS-Bank.
7. The *connectIPS*-Bank will send the notification to the *connectIPS* core engine which will then redirect the response.
   - 7.1. The merchant's portal will receive the response, based on which the further processing will be done by the merchant's application.
   - 7.2. If the debit transaction is successful, it will call the web service exposed for outward direct credit from the *connectIPS*-Bank (customer's bank) that will initiate a direct credit transaction (Pacs08 with special purpose category) in NCHL-IPS system. The transaction details will also be displayed in the NCHL-IPS system for the customer's bank.
   - 7.3. In case the Integration Module of *connectIPS* is not subscribed by the bank, it will then call the web service of the outward direct credit from the third party integration middleware component, which will generate Pacs008 with the specified purpose cateogry. (Bank will have to ensure that such API is exposed for *connectIPS* Core Module for initiating outward direct credit and also provide response for the status of such transaction).
   - 7.4. The purpose category will be ECPG (Guaranteed E-Commerce Payment) for transaction up to the defined limit.
8. The NCHL-IPS system (as a normal process) will forward the transaction to the merchant's bank as inward credit transaction for crediting merchants account.
9. If the Integration Module is also subscribed by the Merchants' bank then it will credit the merchant's account in the CBS after NCHL-IPS settlement is confirmed. Additional configuration will also be available in the inward integration component (of Integration Module) which will be activated for specific purpose category (ECPG) such that it will credit the beneficiary account on real time prior to the NCHL-IPS settlement. For this case the debit cap of the issuing bank (customer's bank) will be utilized and backed by the settlement guarantee mechanism.
10. The final settlement between the banks will be done on deferred net settlement basis.

4. Merchant Interface Description

This section describes the necessary information for integrating merchant e-commerce application/ site with the *connectIPS* Core Module.

4.1. URL: /connectipswebgw/loginpage

4.2. METHOD

POST method.

4.3. PARAMTERS

#	Field Name	Data Type	Length	Presence	Description
1	MERCHANTID	Integer	20	Y	Merchant ID is and unique identifier to identify merchant in the system. Merchant ID will be provided by NCHL upon registering merchant for *connectIPS* Core Module on banks' request.
2	APPID	String	20	Y	Unique identification, which will be used to identify the account details of the merchant's application. A merchant can have multiple applications based on different banks account used for various shopping sites. Application Id will be provided by NCHL after registration.
3	APPNAME	String	30	Y	Application name to identify merchant as well as originating application.
4	TXNID	String	20	Y	Transaction Id which will be used to reconcile transaction between merchant and NCHL. Transaction Id must be unique for each app.
5	TXNDATE	String	10	Y	Transaction Date is the transaction origination date. Date must be in DD-MM-YYYY format.
6	TXNCRNCY	String	3	Y	Currency of transaction. Eg; NPR, USD, GBP etc.
7	TXNAMT	Integer	20	Y	Transaction amount in paisa.
8	REFERENCEID	String	20	Y	Reference Id.
9	REMARKS	String	50	Y	Remarks.
10	PARTICULARS	String	100	Y	Transaction Remarks.
11	TOKEN	String	512	Y	Token for generated transaction details. Hash value must be generated as a token using transaction detail and private key provided by NCHL. Eg; mpqZ3kyEBjhiGKlYMv6OXe4kT8ID5gDr6wRdfd0hAcwlOcKrJn8WHFd5t7V2OCtZrKrEu0BbQeleQbA8kj766PT6J/7eakXFZURn1gedczCovBZq7Hz79lU5KQA58WSCv3sTs3mfY8Qspaz/VbUgHJKNK6thFeNdcs8rNWfXFlfJm9V84Wem2wNC5bjwzd8tZPVHa1BHiF+8eBgOEu8vhvs1tW6VUVbOr/U3ZOZNwaG3ZCL0GUnwF8qrmSoKexj6DDZLZOKB6xMsbTnQCu6i4nn2uGwSmAS3F3kUt5+cjmd4TLURFrYw0UKAXgKlU3tRanhAEN3dOpIisSRaBjwFHQ==

4.4. PROCESS TO GENERATE TOKEN

i. Generate string based on transaction detail in below format.

String message = “MERCHANTID=<Value of MERCHANTID>, APPID=<Value of APPID>, APPNAME=<Value of APPNAME>, TXNID=<Value of TXNID>,TXNDATE=<Value of TXNDATE>,TXNCRNCY=<Value of TXNCRNCY>,TXNAMT=<Value of TXNAMT>,REFERENCEID=<Value of REFERENCEID>,REMARKS=<Value of REMARKS>,PARTICULARS=<Value of PARTICULARS>,TOKEN=TOKEN”

Example:
String message = “MERCHANTID=1,APPID=MER-1-APP-1,APPNAME=Inland Revenue Department,TXNID=8024,TXNDATE=08-10-2017,TXNCRNCY=NPR,TXNAMT=1000,REFERENCEID=1.2.4,REMARKS=123455,PARTICULARS=12345,TOKEN=TOKEN”

ii. Generate hash value of the above string using private key of merchant certificate provided by NCHL.

iii. Send the generate hash value in TOKEN field.

Example:
String TOKEN =
“mpqZ3kyEBjhiGKlYMv6OXe4kT8ID5gDr6wRdfd0hAcwlOcKrJn8WHFd5t7V2OCtZrKrEu0BbQeleQbA8kj766PT6J/7eakXFZURn1gedczCovBZq7Hz79lU5KQA58WSCv3sTs3mfY8Qspaz/VbUgHJKNK6thFeNdcs8rNWfXFlfJm9V84Wem2wNC5bjwzd8tZPVHa1BHiF+8eBgOEu8vhvs1tW6VUVbOr/U3ZOZNwaG3ZCL0GUnwF8qrmSoKexj6DDZLZOKB6xMsbTnQCu6i4nn2uGwSmAS3F3kUt5+cjmd4TLURFrYw0UKAXgKlU3tRanhAEN3dOpIisSRaBjwFHQ==”

4.5. RESPONSE

Response will be provided in respective Success and Failure URL’s, which are configured at the time of registering merchant’s application in *connectIPS* Core Module.

Example Request

```
<form action="https://www.connectips.com/connectipsgw/loginpage" method="post">
        <br>
        MERCHANT ID
        <input type="text" name="MERCHANTID" id="MERCHANTID" value="1"/>
        <br>
        APP ID
        <input type="text" name="APPID" id="APPID" value="MER-1-APP-1"/>
        <br>
        APP NAME
        <input type="text" name="APPNAME" id="APPNAME" value="QFX Movie Ticket"/>
        <br>
        TXN ID
        <input type="text" name="TXNID" id="TXNID" value="120"/>
        <br>
        TXN DATE
        <input type="text" name="TXNDATE" id="TXNDATE" value="13-07-2017"/>
        <br>
        TXN CRNCY
        <input type="text" name="TXNCRNCY" id="TXNCRNCY" value="NPR"/>
        <br>
        TXN AMT
        <input type="text" name="TXNAMT" id="TXNAMT" value="20000"/>
        <br>
        REFERENCE ID
        <input type="text" name="REFERENCEID" id="REFERENCEID" value="REF-001"/>
        <br>
        REMARKS
        <input type="text" name="REMARKS" id="REMARKS" value="RMKS-001"/>
        <br>
        PARTICULARS
        <input type="text" name="PARTICULARS" id="PARTICULARS" value="PART-001"/>
        <br>
        TOKEN
        <input type="text" name="TOKEN" id="TOKEN" value="
mpqZ3kyEBjhiGKlYMv6OXe4kT8ID5gDr6wRdfd0hAcwlOcKrJn8WHFd5t7V2OCtZrKrEu0BbQeleQbA8kj7
66PT6J/7eakXFZURn1gedczCovBZq7Hz79lU5KQA58WSCv3sTs3mfY8Qspaz/VbUgHJKNK6thFeNdcs8r
NWfXFlfJm9V84Wem2wNC5bjwzd8tZPVHa1BHiF+8eBgOEu8vhvs1tW6VUVbOr/U3ZOZNwaG3ZCL0G
UnwF8qrmSoKexj6DDZLZOKB6xMsbTnQCu6i4nn2uGwSmAS3F3kUt5+cjmd4TLURFrYw0UKAXgKlU3t
RanhAEN3dOpIisSRaBjwFHQ=="/>
        <br>
         <input type="submit" value="Submit">
</form>
```

4.6. PAYMENT VALIDATION

REST based API has been provided to validate the status of a transaction. The API requires a basic authentication process which is completed using the provided app id as user name along with password. JSON request must contain merchant id, app id, reference id, transaction amount and token. Reference Id is the TXNID field value supplied during the payment request as described in previous sections. Token is basically a hash value signed with the digital certificate of the creditor.

URL:/api/creditor/validatetxn

Basic Authentication:

User Id: <App Id>
Password: <Password>

Hashing String Format for token:

MERCHANTID=<Merchant Id>,APPID=<App Id>,REFERENCEID=<TXNID in the Request>,TXNAMT=<Transaction Amount>

Example:

Request:
{

```
    "merchantId": 2,
    "appId": "MER-2-APP-1",
    "referenceId": "20176",
    "txnAmt": 50000,
    "token":
"dfjggyRCeG92GSbq6/uhEV9a7eNX17Ij15b+qJDtXYKhOOpXq7WPVznRg2wg8Tm+IG/ay3bCKgOH13Vi
7Yentt9drU5S6ATc4O6uRQwX40cm64NLWzfvImzS7w3LTd5DgosvLlSTed8p1OlQ065P1Wt7nI/uyNMcE
21vniBo1/RHaTeC6dREYYKD00jnrJ/8zGvpVWn/JeOZCeAYCswMS5+g6GPOh+OQxtmxuekrBvYIV57H
ONv+vjc9pLTof3m+3LrB/UVnrxpWoPznSAuYLLF8tkE/GW0OnlCMGFdCPqdKZEUnriLz7Mx1hDjyIad6eu
E2q7vH4XCYj8QghhLdRA=="
}
```

Response:

```
{
    "merchantId": 2,
    "appId": "MER-2-APP-1",
    "referenceId": "20176",
    "txnAmt": "50000",
    "token": null,
    "status": "SUCCESS",
    "statusDesc": "TRANSACTION SUCESSFULL"
}
```

4.7 Tranaction Details

To extract the details of merchant and creditor transaction details, A REST based API is provided as below. The API requires a basic authentication process which is completed using the provided app id as user name along with password. JSON request must contain merchant id, app id, reference id , transaction amount and token. Reference Id is the TXNID field value supplied during the payment request as described in previous sections. Token is basically a hash value signed with the digital certificate of the creditor.

URL: `/api/creditor/gettxndetail`

Basic Authentication:

User Id: <App Id>
Password: <Password>

Hashing String Format for token:

MERCHANTID=<Merchant Id>,APPID=<App Id>,REFERENCEID=<TXNID in the Request>,TXNAMT=<Transaction Amount in paisa>

Example:

Request: Same as that of validatetxn

Response:

```
{
    "status": "SUCCESS",
    "statusDesc": "TRANSACTION SUCESSFULL",
    "merchantId": 11,
    "appId": "WAL-IME-0001",
    "referenceId": "201809271623089187",
    "txnAmt": "100000",
    "token": null,
    "txnId": 9652,
    "txnDate": 1538044712602,
    "txnCrncy": "NPR",
    "chargeAmt": 500,
    "chargeLiability": "MN",
    "refId": "9813033299",
    "remarks": "NCHL-9813033299",
    "particulars": "IMEPAY"
}
```

5. API Specification for revenue payment

In order to extend the government revenue payment through the technical members, a rest base API is provided as below.The API requires a basic authentication process which is completed using the provided user name and password. The basic workflow for the revenue payment will be as below.

- Technical member's customer portal will capture EBPP Number (Request Code) and amount. And it will call our /getebppdetails API.
- Our API will provide detail information about the EBPP Number after confirming it from FCGO API along with the provided amount.
- Technical member will deduct their customer at their side and send confirmation request at our confirmation API.
- System will deducted already registered technical member's account and send credit message to the revenue bank.

Basic Authentication:

User Id: <App Id>

Password: <Password>

POST URL: /api/technical/getebppdetails
This API will provide description about the requested EBPP Number. Amount against the EBPP Number must match the actual amount in the original voucher.
Request Parameters:

Parameter	Description
ebppNumber	Unique number generated in revenue portal for payment tracking.
totalAmount	Total amount specified against the EBPP Number in paisa.
token	Digitally signed token from the private key of API consumer.

Token String: "EBPPNUMBER="+< ebppNumber >+",TOTALAMOUNT="+<totalAmount>

ii. Generate signed hash value of the above string using private key of certificate.
iii. Send the generate hash value in token field.

Example:

```
{
         "ebppNumber": "2075-24253",
         "totalAmount":1800000,
        "token":"D1xAD6zEHvRpAbd6rUUekdAPpc+RvGnaL8bKbKUf9MVKrjrr8zZN2JLer87PM5s0UVlY
        Xh7KvUW8s0GwjAlPTmZkDWr3dIHlwVZqVOCqf23Ji13BjqhrAtwxPOq9Bjtbb3Pe+l4dcuSj6RZiBv
        7SoVkYV0a4BkiGORL8U7lAv62Vi/00pFEDcfibqvtAluRzoDsboJh3+n0tmxai68+UOzPLKPQ0ofg7
        cgoTR3xUYVX8kdkR9FkSRM+os5DIbB1WN21spAo23sRhxS6GX4AYABhYYmwp7+aTAbgc0C4
        VR6epyIfGYKigjmRRXsnVFyevsbBatLiKgh9u5Kd/1kzY9w=="
}
```

Response Parameters:

Parameter	Description
ebppNumber	Unique number generated in revenue portal for payment tracking same as in the request.
txnId	Unique transaction in *connect*IPS against the provided EBPP Number.
totalAmount	Total amount to be paid.
payerEDesc	English description of the payer. This may also contain Nepali Unicode character.
pan	Pan Number of the Payer.
rcAgencyEDesc	Tax agency where the tax is being paid.
status	Status of voucher against the EBPP Number
responseCode	Response Code for the request.
responseDescription	Response Description against the response code.

Example:

```
{
        "ebppNumber": "2075-24253",
        "txnId": 10834,
        "totalAmount": 1800000,
        "payerEDesc": "झापा इन्टरटेनमेन्ट",
        "pan": "601459407",
        "rcAgencyEDesc": "Internal Revenue Office",
        "status": "1",
        "responseCode": "000",
        "responseDescription": "Success"
}
```

POST URL: /api/technical/confirmebpppayment
Payment confirmation API should be provided with the following parameters:
Request Parameters:

Parameter	Description
txnId	Transaction Id in *connect*IPS against the Payment
totalAmount	Total amount to be paid in paisa.
ebppNumber	Unique number generated in revenue portal for payment tracking.
bankId	Debit Bank Id in *connect*IPS.
branchId	Debit Branch Id in *connect*IPS
accountId	Debit account in *connect*IPS
token	Digitally signed token from the private key of API consumer.

Token string: “TXNID=”+<txnId>+”, TOTALAMOUNT =” +<totalAmount>+”,EBPPNUMBER=”+<ebppNumber>+”,BANKID=”+<bankId>+”,BRANCHID=”+<branchId>+”,ACCOUNTID=”+<accountId>

ii. Generate signed hash value of the above string using private key of certificate.
iii. Send the generate hash value in token field.

Example:

```
{
        “txnId”: 10834,
         "totalAmount":1800000,
        "ebppNumber": "2075-24253",
        “bankId”:”1001”,
        “branchId”:”140”,
        “accountId”:”0140256897998”,
        "token":"D1xAD6zEHvRpAbd6rUUekdAPpc+RvGnaL8bKbKUf9MVKrjrr8zZN2JLer87PM5s0UVlY
```

```
            Xh7KvUW8s0GwjAlPTmZkDWr3dIHlwVZqVOCqf23Ji13BjqhrAtwxPOq9Bjtbb3Pe+l4dcuSj6RZiBv
            7SoVkYV0a4BkiGORL8U7lAv62Vi/00pFEDcfibqvtAluRzoDsboJh3+n0tmxai68+UOzPLKPQ0ofg7
            cgoTR3xUYVX8kdkR9FkSRM+os5DIbB1WN21spAo23sRhxS6GX4AYABhYYmwp7+aTAbgc0C4
            VR6epyIfGYKigjmRRXsnVFyevsbBatLiKgh9u5Kd/1kzY9w=="
}
```

Response Parameters:

Parameter	Description
txnId	Unique transaction in *connect*IPS against the provided EBPP Number.
ebppNumber	Unique number generated in revenue portal for payment tracking same as in the request.
totalAmount	Total amount paid in paisa.
responseCode	Response Code for the request.
responseDescription	Response Description against the response code.

Example:

```
{
        "txnId": 10834,
         "ebppNumber": "2075-24253",

           "totalAmount": 1800000,
           "responseCode": "000",
           "responseDescription": "Success"
}
```