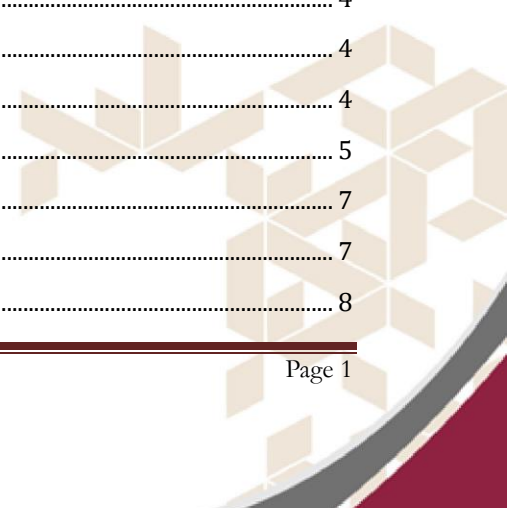




## Corporate API – Process and Requirement Specification

### Contents

A. Objective.....	2
B. Process Flow.....	2
1: Bene Registration (SYNC API).....	2
2: Beneficiary Enquiry (SYNC API).....	3
3: Payment Request Initialization (ASYNC API).....	3
4: Authorization through Internet Module (Optional).....	4
5: Processing of Payment Requests.....	4
6: Call Back API.....	4
7: Invoking Get Status API (SYNC API).....	5
7.1: Exceptions.....	7
7: Balance Enquiry API (SYNC API).....	7
C. Validations.....	8



1. Request Validation.....	8
2. Business Validations.....	8
D. Pre-requisites.....	8
E. Security Considerations.....	10
F. Steps for generating CSR File.....	10
G. Cut-off Timings for Payment Modes.....	11
H. Network Connectivity.....	12
I. Annexure – Fund Transfer, Get Status, Balance Enquiry, API format, API ERROR CODE LIST,.....	12

## A. Objective

This document would basically envisage the scope of requirement, payment request/ Balance enquiry/ get status/ call back and transaction process workflow.

## B. Process Flow

### 1: Bene Registration (SYNC API)

1. Axis bank would expose the API's for Bene Registration.
2. Client would consume the API to initiate the Beneficiary Registration
3. Once the Registration requests reaches Axis Bank's server end, Bank will validate and decrypt all the Registration request.
4. As a best practise, in a single API call corporate should send single beneficiary details in a way that the payload will be lighter so that real time processing and status can be managed
5. Post successful decryption Bank will give response as below with HTTP status as 200 OK to the customer's request.
  - In case of Successful parsing

```
"BeneficiaryRegistrationResponseBody": {
  "data": {
    "beneDetails": [
      {
        "beneMobileNo": "7719871404",
        "beneBankName": "HDFC Bank",
        "beneName": "Sangeetha Mobiles Pvt ltd",
        "beneIfscCode": "HDFC0000523",
        "beneCode": "VEND01264215",
        "beneEmailAddr1": "Bhuvaneshwari@sangeethamobiles.com",
        "corpCode": "DEMOCORP11",
        "status": "ACTIVE",
        "beneAccNum": "5230330001915"
      }
    ], "checksum": " c1ca326533f55d0aa93c1caffde30769a71e3265"
  },
  "message": "Success"
```

```
"status": "S"
}}
```

- In case of any rejections due to format or data validation, a response message at initial level with reason of failure will be given.

```
"BeneficiaryRegistrationResponseBody": {
  "data": "",
  "message": "Invalid Corporate Code",
  "status": "F"
```

Response Code	Description of Response "message": "	Expected Action from customer
200 Status:"S"	Registration Request Received at Bank end.	Enquiry can be invoked
200 Status:"F"	Any form of Business Validation Error	Reinitiate the request with necessary Changes

## **2: Beneficiary Enquiry (SYNC API)**

- Corporate customer can opt for Beneficiary Enquiry Services
- Corporate have the ability to fetch Beneficiary details on the basis of following
  - a. Corpcode – All the beneficiary registered against a corp code will be fetched
  - b. Corpcode + Benecode – Beneficiary details for specific bene-code will be fetched

If the count of Beneficiary is more than 10K, then a .csv file will be triggered after 30 mins to the mail address captured in the API request.

## **3: Payment Request Initialization (ASYNC API)**

1. Axis bank would expose the API for payment initiation.
2. Client would consume the API to initiate the Payment request
3. In this scenario Axis bank acts as API service provider and client who consumes acts as web service consumer.
4. Client who consumes the API Services will send the payment request to Axis bank through internet which would be HTTPS based communication.
5. Once the transaction reaches Axis Bank's server end, Bank will validate and decrypt all the payment request.
6. As a best practise, in a single API call corporate should send single transaction details in a way that the payload will be lighter so that real time transaction processing and status can be managed
7. Post successful decryption Bank will give response as below with HTTP status as 200 OK to the customer's request.

- In case of Successful parsing
 

```
"TransferPaymentResponseBody": {
  "data": "",
  "message": "Success"
  "status": "S"
}
```

- In case of any rejections due to format or data validation, a response message at initial level with reason of failure will be given.

```
"TransferPaymentResponse": {
  "data": "",
  "message": "-> corpAccNum can't be blank.",
  "status": "F"
```

8. In case of any Failure of request parsing failures or internal errors system will send any Non 200 HTTP response.

Response Code	Description of Response "message": "	Expected Action from customer
200 Status:"S"	Transaction Received at Bank end. Awaiting Processing	Enquiry to be invoked after the defined set of time interval covered in Get Status API.
200 Status:"F"	Any form of Business Validation Error	Reinitiate using the same CRN after necessary changes
Any code other than 200	Timeout error, TPS Breach, Unknown Exceptions	Customer should retry the payment request with same CRN

#### **4: Authorization through Internet Module (Optional)**

1. Once the payment requests are received at the bank's end successfully, these records will be available on Power Access/ Corporate Internet Banking / Corporate Mobile App portal for the users to Approve.
2. Transactions will be available to the Users as per the Authorization Matrix which has been setup 3. In case the transactions are "Pre- authorized" the transactions will be processed by system as STP.
4. In case of Salary payment processing, the transactions should be approved on or before the Value Date of the transactions to process - transactions not Approved will be "Expired" by the system and Rejected automatically.

#### **5: Processing of Payment Requests**

1. Authorized transactions are further sent for processing as per the payment mode i.e. RTGS, NEFT, IFT and IMPS.
2. In case of Salary Payments, the transactions will be processed only on the Value Date mentioned in the payment request for the transactions.

Payment Method	Standard TAT (99% txns)
NEFT	60 Mins
RTGS	30 Mins
IMPS	2-5 Mins
IFT	20-30 Mins
UPI	20-30 Mins

#### **6: Call Back API**

1. Customer will host an API which will be consumed by Axis Bank to post txns status reverse feed.
2. Once a transaction is processed, Axis Bank will generate and provide the Reverse feed for the transactions.
3. The reverse feed will be provided with the payment status, status description transaction no. / UTR number etc. as per the defined format.
4. Axis Bank will retry this request until 200 OK is received
5. Refer Callback API document.

Status Code	Description of Status	Expected Action from customer
200 OK	Reverse- feed Posted Successfully	NA
Any code other than 200 / Timeout	Retry will be done by Bank	Corporate to check and resolve the issue.

### 7: Invoking Get Status API (SYNC API)

- Corporate client to use “Get Status” API for inquiring/fetching the payment Status.
- Once a payment initiation has been done, Get Status API can be invoked after following time intervals

Payment Method	Initial Enquiry	Subsequent Enquiry
NEFT	30 Mins	60 Mins
RTGS	5 Mins	30 Mins
IMPS	5 Mins	30 Mins
IFT	5 Mins	30 Mins
UPI	5 Mins	30 Mins

- The customer needs to keep enquiring till they get either of the following status

	<i>transactionStatus</i>	<i>transactionStatus</i>	Action
IMPS	Return	Refer Annexure	No action/Retry Using the same CRN
	Rejected	Refer Annexure	No action/Retry Using the same CRN
	Processed	Successful	No Action
	Pending	Depending on which system it is pending	Status Enquiry to be done till terminal state received
IFT	Rejected	Refer Annexure	No action/Retry Using the same CRN
	Processed	Success	No Action
	Pending	Depending on which system it is pending	Status Enquiry to be done till terminal state received
NEFT/RTGS	Return	Refer Annexure	No action/Retry Using the same CRN
	Rejected	Refer Annexure	No action/Retry Using the same CRN
	Processed	Success	No Action
	Processed	Credited to beneficiary on date time stamp	No Action
	Pending	Depending on which system it is pending	Status Enquiry to be done till terminal state received

- Get Status API will give the current status of transaction i.e. the status at that point of time.
- Please find the link for the following error codes:
  - UPI: <https://www.axisbank.com/docs/default-source/default-document-library/upi-response-codes.pdf>

- IMPS: <https://www.axisbank.com/docs/default-source/default-document-library/imps-response-codes.pdf>
- While doing an enquiry both the combination of response code and transactionStatus needs to be referred as shared in the response API.
- For example:
  - In case of IMPS if "responseCode" is "00" and "transactionStatus" as "PROCESSED", then corporate can mark the transaction as success.
  - In case of IMPS, if "responseCode" is "M0" and "transactionStatus" as "REJECTED", then corporate can reject the transaction
- In case the combination of response code and transaction status shared in the above link does not match with the response received in the API, then the transaction needs to be marked as "PENDING".
- Any response code / status combination not available in the response code sheet to be kept as pending and taken up with service team for confirmation.
- In case of NEFT/RTGS "transactionStatus" as "Processed" and "statusDescription" as "Success" means corporate account has been debited and send to beneficiary bank (This is first level response when transaction is sent to RBI). This status is not terminal & can be changed to RETURN post confirmation from the beneficiary bank.
- There are few Banks which do not provide the credit confirmation and the first Level Response is deemed Success
- "Rejected" status is terminal status and cannot be changed later.
- On Payment Rejected/Returned from Bene Bank:
  - Transaction status will be updated as RETURN with standard rejection/return reason
  - If the transaction is returned from the Beneficiary bank as a fresh inward the status of the original txn will be reflected as "SUCCESS". This need to manually be managed by Corporate.
- The Corporate may retry for the rejected/return transactions using the same CRN.
  - If the previous transaction status is PROCESSED: the client will get a message of Duplicate transaction ID.

```

"CUR_TXN_ENQ": [
  {"corpCode": "DEMOCORP87",
   "statusDescription": "Success",
   "batchNo": "1134",
   "utrNo": "AXISCN0006388931",
   "processingDate": "08-06-2021 15:07:23",
   "responseCode": "ACAR",
   "crn": "ulx13639040796621",
   "transactionStatus": "PROCESSED"
  },
  {"corpCode": "DEMOCORP87",
   "statusDescription": "Duplicate payment. Payment has been already made for the CRN
   <ulx13639040796621> with Bank Txn ID <CN0006388931>.",
   "batchNo": "1136",
   "utrNo": null,
   "processingDate": "08-06-2021 15:47:10",
   "responseCode": "F404",
   "crn": "ulx13639040796621",
   "transactionStatus": "REJECTED"
  }
],
"errorMessage": null,
"checksum": "ea2f25879cfc337b0a7a1e453a44dc85"
}

```

```
"message": "Success",  
"status": "S"}}
```

- If the previous transaction status is REJECTED: the transaction can be taken as fresh and retried at Corporate Client end for the same CRN.
- If Same CRN is invoked multiple times, then the status enquiry will contain all the historic data and corporate to consider the status on the basis of timestamp in the processing date.
  - The Corporate may retry for the rejected/return transactions using the same CRN.
- The NEFT transaction when initiated for normal IFSC UTIBxxxxxxx, then it automatically gets converted to Internal Fund Transfer (IFT) transaction. The handling for such scenarios is as follows:
  - You can mark the IFT/NEFT/RTGS transaction as success if you get response code as '000' and "transactionStatus" as *PROCESSED* ' in both GetStatus API and callback.
  - If you receive any new response code or combination, then it needs to be treated as pending
- The NEFT/IFT/RTGS transaction when initiated for Type C IFSC, it automatically gets converted to NEFT.
  - You can mark the IFT/NEFT/RTGS transaction as success if you get response code as '000' and "transactionStatus" as *PROCESSED* ' in both GetStatus API and callback.
  - If you receive any new response code or combination, then it needs to be treated as pending
  - Please refer to the list of type C IFSC codes



TypeC\_IFSC.xlsx

**NOTE: Axis Bank Stores the CRN for current FY only. Therefore, CRN validation is done only for the current FY.**

### **7.1: Exceptions**

1. Retry Mechanism to be built by client:
  - a. In case there is a TIMEOUT received for the 'Transfer payment' request, then customer should retry the transaction with same CRN.
    - System will process only a single transaction out of the 2 requests, whichever is received first
  - b. In case during the Enquiry request, customer receives status as 'CRN Not Found' then customer should retry the transaction with same CRN.
    - System will process only a single transaction out of the 2 requests
  - c. In case during the Enquiry request, customer receives status as 'CRN Not Found' but customer system does not allow retry of txns then customer should continue enquiry for at least 1 hour. Post this also if status is not updated, then contact Bank customer care. Client should have the option to reject the transaction at their end post verification from the ops team. Suggest reinitiating the transaction with same CRN
  - d. In case during the Enquiry request, customer receives a status a non-standard/ not defined status then customer should contact Bank customer care before reinitiating the transaction

### **7: Balance Enquiry API (SYNC API)**

1. Axis bank would expose the API's for Balance Enquiry.
2. Client would consume the API to enquire the runtime balance
3. Once the request reaches Axis Bank's server end, Bank will validate and decrypt all the request.
4. Post successful decryption Bank will give response as below with HTTP status as 200 OK to the customer's request.
  - In case of Successful parsing

```

"GetAccountBalanceResponseBody": {
  "data": {
    "channelId": "TXB",
    "corpAccNum": "248012910169",
    "corpCode": "DEMOCORP11",
    "Balance": "-454327168887.67",
    "checksum": "d331ca31b40d76879252b9cd021b3215"
  },
  "message": "Success",
  "status": "S"
}
}
}

```

- In case of any rejections due to format or data validation, a response message at initial level with reason of failure will be given.

```

"GetAccountBalanceResponseBody":{
  "data": "",
  "message": "Debit account number incorrect",
  "status": "F"
}}

```

### C. Validations

There are 2 levels of validation of payment request initiated from client end for payment initiation. The details of the same are mentioned as below:

#### 1. Request Validation

All the API's have few mandatory and non-mandatory fields.

- Structure validation, Security validation (Certificate, IP, Channel ID etc.)
- If the Mandatory fields are not present, then request will not process.
- Even if non mandatory fields is not present there will be no validation, these fields will get validated only when they are present in request.

#### 2. Business Validations

In business validation we are considering request should be correct for that we are validating checksum. Another validation will be corporate code and channel id should be mapped properly so that only correct request will get processed.

### D. Pre-requisites

For establishing connectivity between the Customer & Axis Bank production system


#### a. Details to be shared by Bank

- Bank will share API URLs for various Request types
- Bank will share the Signed certificate (.cer) of corporate based on CSR (certificate signing request) of client. This is mandatory for 2-way SSL configuration.
- Bank will share the basic credentials for UAT / Production initiation

#### b. Details to be shared by Client

- Client to share Certificate Signing Request [CSR]
- Client to share Public IP's for both UAT and Production Environment



- 
- **Details to be shared by Client for Reverse-feed (Call back URL)**
  - Client to provide reverse feed end point URL for posting reverse feed which should be HTTPS TLS v1.2 enabled
  - Client to share credentials for accessing End point.
  - Client to share Encryption Algorithm and respective keys

## E. Security Considerations

Following are the security considerations which will need to be followed by the consumers for successful connectivity with the application:

### 1. HTTPS and two way SSL

All consumers will be needed to invoke the application over HTTPS protocol. We also have two way SSL established. This means that we would be validating the consumer certificate. Hence it is required that the consumer has a certificate and the same is shared with us as a prerequisite.

### 2. IP whitelisting

We allow only select IP addresses to access our application over the internet. Hence as a consumer it may be required to whitelist all the ip addresses that the consumer would be consuming the application from. This step is a prerequisite to setup successful connectivity.

### 3. Symmetric encryption

The API to be invoked accepts the request body encrypted using encryption algorithm. The body must be encrypted using AES-128 encryption. The encryption key will be provided and will be different for each consumer.

### 4. Other Parameters

**Client Keys** - All APIs require Client id and secret that is generated as a part of HTTP Header.

**Checksum** – A checksum is a value used to verify the integrity of a file or a data transfer. This is calculated only on attributes within body.

**Channel ID** - Channel ID is client specific which will be configured on API Connect. It will be provided by API Connect to all the clients. It will also be used for decryption of the encrypted request body sent by the client.

## F. Steps for generating CSR File

In the current scenario, corporate customer is the service consumer (client) and Axis Bank is the service provider (server). All the clients consuming our APIs have to follow security guidelines one of which is 2 way SSL.

2-way SSL - This means that we would be validating the consumer certificate. Hence it is required that the consumer sends a CSR to us, we will sign the same and return the signed certificate which consumer needs to configure and produce during each communication. Certificate shared should be valid and should have the usage for client ssl. Consumer needs to invoke the application over HTTPS protocol. We will use TLSv1.2 protocol

Kindly refer below step (Two way ssl )

1) Generate a key

```
openssl genrsa -aes128 -out <keyfile.key>> 2048
```

2)Generate a CSR

```
openssl req -new -key <<keyfile.key>> -out <<csrfile.csr>>
```

3) Axis will sign the CSR and share you the CRT

4) Validate the CRT

```
openssl x509 -in <<certfile>> -text -noout
```

5) Create the cert chain

```
cat <<certfile>> ../SakshamUATIntermediateCertificate.crt ../SakshamUATRootCertificate.crt > Corporate-cert chain.pem
```

6) Create the pkcs12 keystore

```
openssl pkcs12 -export -out <<keystore.p12> -inkey <<keyfile.key>> -in <<certfile>> -name <<aliasname>> certfile  
../SakshamUATRootCertificate.crt -certfile ../SakshamUATIntermediateCertificate.crt
```

### G. Cut-off Timings for Payment Modes

The files can be sent for processing at any time during the day and the data would be processed as per the standard timelines for each payment mode.

Payment Type	Processing Time	Amount Limit
RTGS*	7 AM to 9.30 PM	Max – No Limit, Min: Rs. 2 lacs
NEFT*	7 AM to 9.30 PM	Max – No Limit, Min: Rs 1.00
FT	7 AM to 11 PM	Max – No Limit, Min: Rs 1.00
IMPS	24x7	Max: Rs.5 lacs, Min: Rs 1.00
UPI	24x7	Max: Rs. 1 lac, Min: Rs 1.00

**\*RBI working days - Transactions approved post cut of will be processed on next working day only. Amount above 1Cr. Initiated after 6 PM will be processed with a delay in TAT due to Third Level Authorization. The TAT covered in this document will not apply to such transaction**

## H. Network Connectivity

For establishing connectivity between the Customer & Axis Bank production system

- Customer's PUBLIC IP – this is required of the server from where the payment request will be sent to bank for processing. And of server to which Bank has to send reverse feed (call back API)
- Customer to whitelist public IPs of Bank. *PROD IPs of Bank will be shared before production.*

	UAT URL	PROD URL
Get Status	<a href="https://sakshamuat.axisbank.co.in/gateway/api/txb/v1/acct-recon/get-status">https://sakshamuat.axisbank.co.in/gateway/api/txb/v1/acct-recon/get-status</a>	<a href="https://saksham.axisbank.co.in/gateway/api/txb/v1/acct-recon/get-status">https://saksham.axisbank.co.in/gateway/api/txb/v1/acct-recon/get-status</a>
Get Balance	<a href="https://sakshamuat.axisbank.co.in/gateway/api/txb/v1/acct-recon/get-balance">https://sakshamuat.axisbank.co.in/gateway/api/txb/v1/acct-recon/get-balance</a>	<a href="https://saksham.axisbank.co.in/gateway/api/txb/v1/acct-recon/get-balance">https://saksham.axisbank.co.in/gateway/api/txb/v1/acct-recon/get-balance</a>
Bene Registration	<a href="https://sakshamuat.axisbank.co.in/gateway/api/txb/v1/payee-mgmt/beneficiaryregistration">https://sakshamuat.axisbank.co.in/gateway/api/txb/v1/payee-mgmt/beneficiaryregistration</a>	<a href="https://saksham.axisbank.co.in/gateway/api/txb/v1/payee-mgmt/beneficiary-registration">https://saksham.axisbank.co.in/gateway/api/txb/v1/payee-mgmt/beneficiary-registration</a>
Bene Enquiry	<a href="https://sakshamuat.axisbank.co.in/gateway/api/txb/v1/payee-mgmt/beneficiary-enquiry">https://sakshamuat.axisbank.co.in/gateway/api/txb/v1/payee-mgmt/beneficiary-enquiry</a>	<a href="https://saksham.axisbank.co.in/gateway/api/txb/v1/payee-mgmt/beneficiary-enquiry">https://saksham.axisbank.co.in/gateway/api/txb/v1/payee-mgmt/beneficiary-enquiry</a>
Fund Transfer	<a href="https://sakshamuat.axisbank.co.in/gateway/api/txb/v1/payments/transfer-payment">https://sakshamuat.axisbank.co.in/gateway/api/txb/v1/payments/transfer-payment</a>	<a href="https://saksham.axisbank.co.in/gateway/api/txb/v1/payments/transfer-payment">https://saksham.axisbank.co.in/gateway/api/txb/v1/payments/transfer-payment</a>

UAT PUBLIC IP address	Port Number
1) 115.112.85.180 2) 36.255.28.180	443

Command to check Telnet	telnet <hostname> <port>  telnet 115.112.85.180 443 telnet 36.255.28.180 443
-------------------------	---

## I. Annexure –

 Fund Transfer & Get Status API_02.06	 Bene Management 20052021.pdf	 Bene Enquiry 20052021.pdf	 FundTransfer Response Codes.xls
---	---	--	--



 Checksum Logic (2).docx	 Symmetric EncryptionDecryptio		
---	---	--	--