AU Small Finance Bank

API Banking Integration Document

Payout API

Enquiry API

Project – API Banking
Document Type – Integration Specification Document.

Prepared by	Reviewed by	Approved by
Kunal Boriwal	Anil Kumar Neelam	
	Sukha Ram	

Change History -

Version	Changed by	Date	Change Information.
1.0			

Contents

Introduction:

The world is evolving with newer technologies at fast-moving speeds and the banking industry is often the first to jump in to experiment with this. Among them, API has been the biggest advancement in the banking segment. Open banking enables Corporates/SMEs, to access consumer's banking and other financial data by using open APIs.

AU Small Finance Bank API Banking enables its partner organizations to seamlessly co-create unique client solutions, enhancing their productivity. API Banking enables to manage your relevant banking activities from your server (such as collections, payments, reconciliations, account balance and retrieving bank statements). That's not all, it links appropriate APIs with your server based on your business needs to give you an integrated banking solution thus saving time on bank branch visits.

Benefits:

1. Bank reconciliation that is error-free and automated.
2. Single and bulk payments are made directly.
3. Verify beneficiary before Payment.
4. End mile transaction status.
5. Data transmitted between the bank and your servers is encrypted.

List of APIs services:

1. Payout API
2. Enquiry API

Payout API process:

- 1st Step -Token Generation - Calling Token OAuth generation service.
- 2nd Step - Payment Posting- Initiating payment creation service.
- 3rd Step - Payment Enquiry - Payment enquiry service API to be used for getting the final status of transaction. It is recommended to schedule the Enquiry API in every 15 minutes.

  Note: NEFT transactions are processed in batches through RBI. It may take 1 to 2 hours to reflect the final status.

Encryption – Decryption Methodology:

- Follow the Symmetric Encryption AES 256 bits -CBC (64 bytes key).

Token OAuth generation service.

Generate Access token by calling URL and select GET method (Query string) –

https://api.aubankuat.in/oauth/accesstoken?grant_type=client_credentials

Refer the sample response -

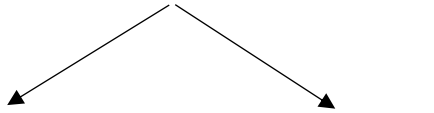

```
{
   "refresh_token_expires_in": "0",
   "api_product_list": "[LDAP, Oauth, Payment, Customer Onboarding, karza,
CBSMiniStatementService, test]",
   "api_product_list_json": [
      "LDAP",
      "Oauth",
      "Payment",
      "Customer Onboarding",
      "karza",
      "CBSMiniStatementService",
      "test"
   ],
   "organization_name": "au-apigee-nprod",
   "developer.email": kunal.boriwal@aubank.in,
   "token_type": "BearerToken",
   "issued_at": "1704950669618",
   "client_id": "2I7UVNalTfFBxm3ZYxOtzYXwXX1PMIJCSSFf6AMipK0H0zR9",
   "access_token": "lEbnG39cJwC4lKUe5fliVA9HFcyR",
   "application_name": "f0556c9d-6c97-40aa-8d4e-c6bb190ef2ce",
   "scope": "",
   "expires_in": "86399",
   "refresh_count": "0",
   "status": "approved"
}
```


***Note: Token will valid upto 24hrs in UAT and 06 months in production.*

1. CNB Payout Services-

1. CNB payment service is used to initiate Internal Fund Transfer / NEFT/RTGS/IMPS transactions (Single +Bulk payment) through CIB platform.
2. Bulk payment up to 50 transactions i.e. nested with multiple NEFT, RTGS, IMPS, Internal Fund Transfer transaction and Batch wise processing is done over here.

Payout service is broadly divided into 2 parts –



Endpoint URL–

Auth URL (UAT) - https://api.aubankuat.in/oauth/accesstoken?grant_type=client_credentials

CNB Payment URL (UAT) - https://api.aubankuat.in/CNBPaymentService/paymentCreation

***Note: Production urls & details will be shared, post UAT sign-off.*

Steps to call API -

Step 1: Generate Access Token through Oauth API (Service details provided below) -> Copy the Access token from the response. Token will valid upto 24hrs in UAT
Step 2: Payload should be encrypted with AES-256 encryption, key will be provided by Bank team.
Step 3: Encrypted payload pass in JSON payload as the request Body (format mentioned below).

Request Payload Input Parameter

Field Name	Data Type	Length	Mandatory	Provided by AU Bank	Description
uniqueRequestId	String	20	M	No	unique Request/Reference number of calling app for Batch request
corporateCode(CIF number)	String	20	M	Yes	corporate code maintained in Core i.e **CIF Number**
corporateProductCode	String	50	M	Yes	corporate product code maintained in CIB Corporate Product can be different from maintained Product.
paymentMethodName	String	50	M	No	Payment Method Name should be same as maintain in corporate product. For different payment modes – **NEFT**– For NEFT txns. **RTGS**– For RTGS txns. **IMPS**– For IMPS txns. **Internal Fund Transfer**– For Internal

					Fund Transfer txns in AUSFB
remitterAccountNo	String	35	M	Yes	remitter Account No
Amount	String	14,2	M	No	payable amount
ifscCode	String	50	M	Yes	Beneficiary ifscCode
payableCurrency	String	20	M	No	payableCurrency Will always be sent as **INR**
beneAccNo	String	35	M	Yes	Beneficiary Account No/ Payment Receiver Account number.
beneCode	String	200	O	No	beneCode
beneName	String	200	M	No	Beneficiary Name/ Payment Receiver name.
valueDate	String	08	O	No	Value date on which transaction will be executed unless value date is behind application date. Back Dated Transaction will be processed on same date. Future Date will be scheduled. Default value will be current date. If no data passed, then it will be considered as Current date
Remarks	String	40	O	No	Optional field to be input while initiating the request.
transactionRefNo	String	25	M	No	This will be unique reference no. for each transaction detail at batch level
paymentInstruction	String	314	M	No	User can input the desirable narration.
email	String	50	O	No	email
phoneNo	String	200	O	No	phoneNo

Response Output Parameter

Field Name	Data Type	Length	Mandatory	Description
uniqueRequestId	String	-	M	unique Request/Reference number of calling app for Batch request
batchNo	String	-	M	unique Batch number
ResponseCode		2	M	0 or 99
ResponseMessage		2	M	Success Failure
ExtendedErrorDetails		20	M	error Details
errorcode		2	M	0 or 99
errormessage		240	M	in case of success Success message other failure message

Refer the Sample Request – Response payload:

1. **INTERNAL FUND TRANSFER Transaction packet:**

Success Request packet	Success Response packet
{ "TransactionCreationRequest": { "uniqueRequestId": "Test021", "corporateCode": "29595833", "corporateProductCode": "ALLPAYMENTSWOA", "transactionDetails": [{ "paymentMethodName": "Internal Fund Transfer", "remitterAccountNo": "2401201151784662", "amount": "1", "ifscCode": "", "payableCurrency": "INR", "beneAccNo": "2302201151638241", "beneCode": "11111", "valueDate": " ", "beneName": "APITEST01", "remarks": "H2H API", "transactionRefNo": "APITEST01", "paymentInstruction": "Payment Instruction 98", "email": "test@gmail.com", "phoneNo": "999999999" }] } }	{"TransactionCreationResponse": { "uniqueRequestId": "Test021", "responseCode": "00", "responseMessage": "ACCEPTED", "batchNo": "01070224018" }}

2. IMPS Transaction packet

Success Request packet	Success Response packet
{ "TransactionCreationRequest": { "uniqueRequestId": "Test01-0", "corporateCode": "29595833", "corporateProductCode": "ALLPAYMENTSWOA", "transactionDetails": [{ "paymentMethodName": "IMPS", "remitterAccountNo": "2401201151784662", "amount": "5", "ifscCode": "KARB0000146", "payableCurrency": "INR", "beneAccNo": "14625001012425600", "beneCode": "11111", "valueDate": " ", "beneName": "APITEST01", "remarks": "H2H API", "transactionRefNo": "APITEST01", "paymentInstruction": "Payment Instruction 98", "email": "test@gmail.com", "phoneNo": "999999999" }] } }	{"TransactionCreationResponse": { "uniqueRequestId": "Test01-0", "responseCode": "00", "responseMessage": "ACCEPTED", "batchNo": "01070224012" }}

Sample Failure packet of Payment transaction

Failure request	Failure Response
{ "TransactionCreationRequest": { "uniqueRequestId": "10301202401", "corporateCode": "28954953", "corporateProductCode": "1211", "transactionDetails": [{ "paymentMethodName": "IMPS", "remitterAccountNo": "2301221548647148", "amount": "121.00", "ifscCode": "DNSB0000001",	{"TransactionCreationResponse": { "uniqueRequestId": "10301202401", "responseCode": "00", "responseMessage": "ACCEPTED", "batchNo": "01260324013" }}

"payableCurrency": "INR", "beneAccNo": "123456041", "beneCode": "11111", "valueDate": "", "beneName": "Palwit1", "remarks": "H2H API", "transactionRefNo": "189_27", "paymentInstruction": "Payment Instruction 98", "email": "abc@gmail.com", "phoneNo": "9999999999" }] } }	

3. **NEFT Transaction Packet:**

Request	**Response**

{
"TransactionCreationRequest": {
"uniqueRequestId": "Test009",
"corporateCode": "29595833",
"corporateProductCode": "ALLPAYMENTSWOA",
"transactionDetails": [
{
"paymentMethodName": "NEFT",
"remitterAccountNo": "2401201151784662",
"amount": "1",
"ifscCode": "KARB0000146",
"payableCurrency": "INR",
"beneAccNo": "14625001012425600",
"beneCode": "11111",
"valueDate": " ",
"beneName": "APITEST01",
"remarks": "H2H API",
"transactionRefNo": "APITEST01",
"paymentInstruction": "Payment Instruction 98",
"email": "test@gmail.com",
"phoneNo": "999999999"
}
]
}
}
	{"TransactionCreationResponse": {
	"uniqueRequestId": "Test009",
	"responseCode": "00",
	"responseMessage": "ACCEPTED",
	"batchNo": "01070224009"
	}}

4. **RTGS Transaction Packet:**

Request	Response

<pre>{
  "TransactionCreationRequest": {
  "uniqueRequestId": "Test008",
  "corporateCode": "29595833",
  "corporateProductCode": "ALLPAYMENTSWOA",
  "transactionDetails": [
  {
  "paymentMethodName": "RTGS",
  "remitterAccountNo": "2401201151784662",
  "amount": "200001",
  "ifscCode": "KARB0000146",
  "payableCurrency": "INR",
  "beneAccNo": "14625001012425600",
  "beneCode": "11111",
  "valueDate": " ",
  "beneName": "APITEST01",
  "remarks": "H2H API",
  "transactionRefNo": "APITEST01",
  "paymentInstruction": "Payment Instruction 98",
  "email": "test@gmail.com",
  "phoneNo": "999999999"
  }
  ]
  }
}</pre>
	<pre>{"TransactionCreationResponse": {
	  "uniqueRequestId": "Test008",
	  "responseCode": "00",
	  "responseMessage": "ACCEPTED",
	  "batchNo": "01060224074"
	}}</pre>

Error code handling -

Error Code	Description	Message	Actionable
200	Success	Success response	
400	Bad request	Bad request	
401	Unauthorized	Invalid authentication credentials	
404	Not Found	Not Found	

500	Internal Server Error	Something went wrong	
503	Service Unavailable	Service Unavailable	
504	Gateway timeout	Gateway timeout	

2. CNB Enquiry Service –

This service is used to check the transaction status of payment initiated through payout service.

End point URL -

UAT Payment Enquiry - https://api.aubankuat.in/CNBPaymentService/paymentEnquiry

Enquiry stages-

Level 1: "status": "OPEN.
Level 2: "status": "TOBEPROCESSED" or "SENTTOCORE"
Level 3: "status": "PAID", with "utr": "N18823200919855" amount will be credited to bene Bank.

Request Input Parameter

Field Name	Data Type	Length	Mandatory	Provided by Bank	Description
uniqueRequestId	String	20	M	No	API unique Request ID for calling enquiry
corporateCode	String	20	M	Yes	CIF maintained in core
batchNo	String	20	M	Yes	Batch number sent by CIB in response in payment initiation call
transactionRequestId	String	20	M	No	Should be passed same as uniqueRequestId from Transaction API Call
transactionRefNo	String	25	M	No	Transaction Reference number send in request on payment initiation call at transaction level. If transactionRefNo is mentioned, then batchNo or transactionRequestId is mandatory

Response Output Parameter

Field Name	Data Type	Length	Mandatory	Description
uniqueRequestId	String	20	M	unique Request Id for enquiry request
batchNo	String	10	O	unique Batch number
ResponseCode	String	2	O	0 or 99
ResponseMessage	String	2	O	Success Failure
ExtendedErrorDetails	String	20	O	error Details
errorcode	String	2	O	0 or 99
errormessage	String	240	O	in case of success, Success message other failure message
transactionRequestId	String	10	O	reference number at batch level from posting request
batchNo	String	10	O	batch number generated by CIB, and sent back to client in response of payment Post request
noOfTransactions	String	10	O	no. of transaction in batch
batchStatus	String	10	O	Batch Status
batchStatusCode	String	10	O	Batch Status Code
beneficiaryName	String	120	O	Beneficiary Name
beneficiaryAccount	String	20	O	Beneficiary Account number
amount	String	14,2	O	transaction amount
paymentMethod	String	05	O	User define – NEFT, RTGS, IMPS, Internal Fund Transfer.
remitterAccountNo	String	20	O	remitter AccountNo
transactionRefNo	String	10	O	transactionRefNo received from client in Transaction Post request
valueDate	String	08	O	value date mentioned in request of Payment Post
payableCurrency	String	14,2	O	payableCurrency
utr	String		M	Unique Ref No. generated by Bank Below is the sample values – RTGS – AUBLR22024020601864248 NEFT – N067242155532001

				IMPS – 403811659667
statusCode	String		M	Status code of transaction
status	String		M	Status of the transaction
errorCode	String		O	Transaction failure code
errorMessage	String		O	Transaction Failure reason

Enquiry based on Batch number

Request payload	**Response payload**

{
"TransactionEnquiryRequest": {
"uniqueRequestId": "5445",
"transactionRequestId": "",
"batchNo": "01060224074",
"transactionRefNo": "",
"corporateCode": "29595833"
}
}
	{"TransactionEnquiryResponse": {
	"uniqueRequestId": "5445",
	"responseCode": "00",
	"responseMessage": "SUCCESS",
	"transactionRequestId": "Test008",
	"batchNo": "01060224074",
	"batchStatus": "SUCCESS",
	"batchStatusCode": "00",
	"noOfTransactions": 1,
	"transactionEnquiryDetails": [{
	"beneficiaryName": "APITEST01",
	"beneficiaryAccount": "14625001012425600",
	"amount": "200001",
	"paymentMethodName": "RTGS",
	"remitterAccountNo": "2401201151784662",
	"transactionRefNo": "APITEST01",
	"valueDate": "2024-02-06",
	"payableCurrency": "INR",
	"statusCode": "101",
	"status": "SENTTOCORE",
	"errorCode": "",
	"errorMessage": "",
	"utr": "AUBLR22024020601864248"
	}]
	}}

Enquiry based on Transaction Request id (Used in case when Batch number is not received)

Request Payload	**Response Payload**

{
"TransactionEnquiryRequest": {
"uniqueRequestId": "5445",
"transactionRequestId": "Test008",
"batchNo": "",
	{"TransactionEnquiryResponse": {
	"uniqueRequestId": "5445",
	"responseCode": "00",
	"responseMessage": "SUCCESS",
	"transactionRequestId": "Test008",

"transactionRefNo": "", "corporateCode": "29595833" } }	"batchNo": "01060224074", "batchStatus": "SUCCESS", "batchStatusCode": "00", "noOfTransactions": 1, "transactionEnquiryDetails": [{ "beneficiaryName": "pranjal", "beneficiaryAccount": "14625001012425600", "amount": "200001", "paymentMethodName": "RTGS", "remitterAccountNo": "2401201151784662", "transactionRefNo": "APITEST01", "valueDate": "2024-02-06", "payableCurrency": "INR", "statusCode": "00", "status": "PAID", "errorCode": "", "errorMessage": "", "utr": "AUBLR22024020601864248" }] }}

Sample Failure packet of Status Enquiry API

Failure request	**Failure Response**

{
"TransactionEnquiryRequest": {
"uniqueRequestId": "TEst01",
"transactionRequestId": "",
"batchNo": "01260324013",
"transactionRefNo": "",
"corporateCode": "28954953"
}
}
	{"TransactionEnquiryResponse": {
	"uniqueRequestId": "TEst01",
	"responseCode": "00",
	"responseMessage": "SUCCESS",
	"transactionRequestId": "10301202401",
	"batchNo": "01260324013",
	"batchStatus": "FAILED",
	"batchStatusCode": "99",
	"noOfTransactions": 0,
	"errorMessage": "Duplicate file uploaded."
	}}

Description of status code Tag (transaction level)

BatchStatus Code	Batch Status
00	SUCCESS
99	FAILED
100	PROCESSING
101	PROCESSING

Note: Processing status means transaction is still in the system and final status is not updated. For processing cases, user should not reinitiate the payment transaction again with same details. Enquiry API to be retrigged for such cases to get updated status.

Error code handling

Error Code	Description	Message	Actionable
401	Unauthorized	Invalid authentication credentials	
404	Not Found	Not Found	
500	Internal Server Error	Something went wrong	

*****Generic test cases / Test data***

1. Beneficiary Details for RTGS/NEFT Payment method

Beni Account	Beni IFSC code	Mobile Num	Email	Beni Name
14625001012425600	KARB0000146	999999999	test@gmail.com	APITEST01

2. Beneficiary Details for IMPS Payment method

Beni Account	Beni IFSC code	Mobile Num	Email	Beni Name
123456041	DNSB0000001	999999999	test@gmail.com	APITEST02

3. Internal Fund Transfer Beneficiary

Beni Account	Beni IFSC code	Mobile Num	Email	Beni Name
2302201151638	NA	999999999	test@gmail.com	APITEST03

S.No	Case	Request	Response	Remarks if any (mark NA if not applicable)

1	Payment initiation request -Success case with single transaction -Internal Fund Transfer			
2	Payment initiation request -Success case with single transaction – NEFT			
3	Payment initiation request -Success case with single transaction – IMPS			
4	Payment initiation request -Success case with single transaction – RTGS			
5	Payment initiation request -Success case with multiple transactions if applicable for corporate in single API			
6	Payment initiation request -Failure case (Ex. Blank BIC code / Missing mandatory req parameter)			
7	Payment initiation request -pending case (ESB should not response to client in defined time)			
8	Payment enquiry request- Success case – basis of batch no			
9	Payment enquiry request- Success case – basis of reference ID			
10	Payment enquiry request- failure case – basis of limit reason			

Authentication Methods:

1. For secure access & seamless experience, we do follow OAuth 2.0 authenticated method prior to every API services call.
2. During the onboarding process, AU Bank will share the credential file (Contract note) separate for both UAT & Prod.
3. Client id, Client secret & encryption key is mentioned over here. This would be primary level of authentication mode.
4. Partner needs to pass these values for end-to-end integration testing.

Procedure to call the OAuth 2.0 Access Token API:

Step #1: Generate Access token by calling URL –
https://api.aubankuat.in/oauth/accesstoken?grant_type=client_credentials
Authorization Type as basic set the Username and password.

#Sample case

Username	Vit8FwGcDwvRJzq6eHoP9G6XFA7LmtTr6eABGbaQ33N1Ht3H
Password	eMRE4ldU2goAr2uZvm5XeRnbQki0WDTgvPXQpQyHJvJeaFi8Ip4TAghI9wBa7PxA

** Note: Username and password will be available as client id and client secret.

Response: Copy the Access Token Field from the response payload

Step #2: To Make respective Service call, Copy the endpoint URL from contract note-
URL : *******
Header: Key-Authentication
Value-<<Bearer Token>>

Body: JSon request Body with encrypted payload as value

Format:
{
"encvalue":"<<response from aesencrypt>>"
}

Response: Copy the response - which will be in encrypted format

Endpoints and Request/Response Formats:

1. Each API services is having specific endpoint urls which is mentioned against each API service description.
2. All services payload is defined in Rest based API- Json format payload.
3. OAuth Token generation, Penny drop services would make use of GET method as query parameter.
4. It is advisable to initiate the integration testing i.e. 1st level through postman tool then, followed with application-level testing.

Error Handling

1. Each API services is defined with appropriate response i.e. Success or failure.
2. In case of failure or error encounter- various error code/ messages are mentioned in each respective service content.
3. This would be helpful in troubleshooting & debugging the issue and getting the expected response.
4. For feasible analysis, you must share the plain text payload with proper timestamp and error screenshot.

Data Security Measures

1. For payout services, penny drop service, Mini statement service, emandate service- bank follows encryption – Decryption methodology.
2. Algorithm keyword: Symmetric encryption AES – 256 bits -CBC – 64 bytes key

Testing and Sandbox Environment:

1. 1st level – Partner needs to execute the end-to-end testing in UAT through postman tool and then, follow with application-level testing. Post result Partner needs to share the Test results with bank.
2. Post test-Results reviewing, we will be considered the UAT sign off.
3. Once UAT is cleared, then partner will be moved to production environment.

Troubleshooting and Support:

1. Partner needs to share the error encountered plain text request and response payload, IP address, API key in notepad for debugging over mail.
2. If the issue persists again, we can connect jointly over MS Teams to close the issue.

Important Notes:

1. Post receiving the prerequisite details, UAT integration details will be shared within 4 working days.

2. For any integration issue, please check the "Troubleshooting and Support" section of the document. For any further query please write a mail to **api.dev-team@aubank.in, CIB_TECH_TEAM@aubank.in.**

3. Client needs to test all the test case scenarios mentioned above and provide signoff to proceed with Production deployment.

4. Bulk testing should be informed prior initiating. It's estimated count should match the values mentioned while sharing the onboarding details i.e. monthly count and avg. value per transactions. Retry mechanism for enquiry service should be discussed with IT team.

5. Post receiving UAT signoff, Bank will require 4- 5 Working days for production deployment and sharing production contract notes. Production movement slot can be raised either on Monday or Thursday in week and next day will confirm and share the contract note.

6. There will be a deployment freeze on every month end from 25th to 4th day of next month.

7. For any queries on the transaction status post go live, please write an email to customercare@aubank.in with transaction details.

Frequently Asked Questions (FAQs)

1. Is it necessary to the Encrypt the payload?
   Ans: Yes, Encryption of payload is mandatory.

2. What kind of Encryption- decryption mechanism followed for Payment and Enquiry API?
   Ans: Encryption AES – 256 bits -CBC – Algorithm.

3. What are several payment modes available?
   Ans: Payment mode available - NEFT, RTGS, IMPS, Internal Fund Transfer.

4. How to check the status of transaction?
   Ans: Final status of payments can be checked by using Payment Enquiry API

5. Will transactions be implemented 24x7 for clients?
   Ans: Yes, transaction can be implemented 24x7, however from 06:30 PM to 12:00 AM, bank will limit the consolidated transaction value upto Rs 1 cr.

6. Whom shall we contact in case of any issue in UAT?
   Ans: Please drop the mail to api.dev-team@aubank.in , looping RM in CC.

7. Is Beneficiary addition being mandatory while initiating Payment API request?
   Ans: Transaction can be done without adding of beneficiary.

8. What is the timing recommended for Payment Enquiry API to use?
   Ans: It is recommended to call Payment enquiry API in fixed frequency of 15 to 30 minutes.

9. What is the ideal response time for NEFT transaction?
   Ans: NEFT transactions are processed in batches through RBI. It may take 30 min to 2 hours to reflect the final status.

10. What is the ideal response time for RTGS/IMPS transactions?
    Ans: Response is real-time.

11. What is Internal Fund Transfer (IFT)? What Transaction Type & IFSC code need to be used in this case?
    Ans: Internal Fund Transfer is transaction done within AU Small Finance Bank where payer & payee account is within the Bank. In this case payment method name will be passed as “Internal Fund Transfer” without IFSC code.