ICICI Bank

API Documentation for Composite Validation

Table of Contents

COMPOSITE VALIDATION API

Introduction

In today's digital era, numerous E-commerce companies, Payment Gateway Aggregators, NBFCs, and Insurance companies are transitioning towards API-based payment solutions, moving away from manual intervention and complex payment processing methods. The banking Industry has witnessed a significant leap forward with the API's. In response to the evolving needs of corporates and after studying the corporate need and its requirement, "Composite Validation API" has been designed thoughtfully.

The Composite Validation API will be used for beneficiary validation check. It is designed to enhance our product services significantly. This API includes two advanced features in IMPS such as Name Matching API and Name Inquiry API and one under UPI that is UPI Validation API which will greatly contribute to the accuracy and efficiency of our processes. The Composite Validation API request and response to Merchant is secured using the encryption and decryption process.

The Composite Validation APIs are hosted on the ICICI Bank Gateway. ICICI Bank Gateway will route any request of the validation API to the respective system of the payment mode. ICICI Bank provide separate integration environment for UAT & production environment testing. In the UAT environment, the client will be able to test all possible scenarios & post confirmation; the client will be moved to Production/Live environment.

Endpoints (URL)

API UAT: https://apibankingonesandbox.icicibank.com/api/v1/composite-validation

API LIVE: https://apibankingone.icicibank.com/api/v1/composite-validation

IMPS

Immediate Payment Service [IMPS], is a service which allows fund transfers within banks across India, in a cost & time efficient manner. IMPS is supported for >700 financial institutions in India.

1. IMPS Name Match API:

The IMPS Name match API with percentage match of Beneficiary Name provides the percentage match of the beneficiary name received in the response against the Beneficiary name given in request packet.

Input parameters

Header			
Input Parameter	Description	Provided by ICICI	Mandatory value
apikey	'API Key' is a unique value generated for each client. API Key will be provided to the client during onboarding and same needs to be used for all further transactions. Client must ensure that requests are being sent from whitelisted IP's only.	Yes	Mandatory
x-priority	This value states the priority order for the mode of payment. For IMPS, x-priority = "0100"	Yes	Mandatory

Body			
Input parameter & type (maximum character)	Description	Provided by ICICI (Yes/No)	Mandatory value
LocalTxnDtTime String(20)	Transaction Date & Time (YYYYMMDDHHmmss)	No	Mandatory

BeneAccNo [String (30)]	Beneficiary Account Number	No	Mandatory
BeneIFSC [String (11)]	Beneficiary IFSC Code	No	Mandatory
TranRefNo [String(50)]	Transaction Reference Number that uniquely identifies the transaction. Unique number should be entered by the client.	No	Mandatory
Amount [String(14 Max LENGTH,2 Min)]	Transaction amount	No	Mandatory
PaymentRef [String (50)]	Transaction remarks, these details will be sent to NPCI. This filed will be reflected in the bank statement also.	No	Mandatory
RemMobile [string (12)]	Remitter Mobile number	No	Mandatory
RetailerCode [String (10)]	Value of “rcode” to be passed	Yes	Mandatory
BcID [String(50)]	Client BCID. This will be provided by ICICI Bank during onboarding.	Yes	Mandatory
PassCode [String(50)]	PassCode gets generated through IMPS and provided by ICICI Bank during onboarding.	Yes	Mandatory
Beneficiary_Name [String (50)]	Beneficiary Name	No	Mandatory
RemName [String (50)]	Sender (remitter) Name	No	Mandatory

A Sample request packet for IMPS Name Match	
Header	apikey = '----------' x-priority = '0100'

Body	{
	"LocalTxnDtTime": "20240216183100",
	"BeneAccNo": "123456041",
	"BeneIFSC": "NPCI0000001",
	"TranRefNo": "20240216183100",
	"Amount": "1",
	"PaymentRef": "IMPSTransfer",
	"RemMobile": "9999988888",
	"RetailerCode": "rcode",
	"PassCode": "447c4524c9074b8c97e3a3c40ca7458d",
	"BcID": "IBCKer00055",
	"Beneficiary_Name": "Kiran Kumar",
	"RemName": "Pratik Mundhe"
	}

Output parameters

Output parameter & type	Description
ActCode [String (4)]	ActCode Value
Response [String (255)]	ActCode Description explains the reason for success, failure or pendency of the transaction
BankRRN [Number (12)]	It is a unique 12-digit number used to refer a particular transaction
BeneName [String (100)]	Beneficiary Name
Match_% [String (5)]	Match percent

A Sample response packet for IMPS Name Match	
Success	Failure

{
"ActCode": "0",
"Response": "Transaction Successful",
"BankRRN": "404718630776",
"BeneName": "Kiran Kumar",
"Match_%": "100.0"
}
	{
	"ActCode": "62",
	"Response": "Reason Unknown",
	"BankRRN": "406512367537",
	"BeneName": "",
	“TranRefNo”:202402211151558
	}

2. IMPS Name Inquiry API:

The API can be used for knowing the bene name without actual money credit into bene account.

Input parameters

Header			
Input Parameter	Description	Provided by ICICI	Mandatory value
apikey	'API Key' is a unique value generated for each client. API Key will be provided to the client during onboarding and same needs to be used for all further transactions. Client must ensure that requests are being sent from whitelisted IP's only.	Yes	Mandatory
x-priority	This value states the priority order for the mode of payment. For IMPS, x-priority = "0010"	Yes	Mandatory

Body			
Input parameter & type (maximum character)	Description	Provided by ICICI (Yes/No)	Mandatory value
BeneAccNo [String (30)]	Beneficiary Account Number	No	Mandatory
BeneIFSC [String (11)]	Beneficiary IFSC Code	No	Mandatory
TranRefNo [String (50)]	Transaction Reference Number that uniquely identifies the transaction. Unique number should be entered by the client.	No	Mandatory

PaymentRef [String (50)]	Transaction remarks, these details will be sent to NPCI. This filed will be reflected in the bank statement also.	No	Mandatory
RemName [String (50)]	Sender (Remitter) Name	No	Mandatory
RemMobile [String (12)]	Remitter Mobile number	No	Mandatory
RetailerCode [String(50)]	Value of “rcode” to be passed	Yes	Mandatory
PassCode [String(50)]	PassCode gets generated through IMPS and provided by ICICI Bank during onboarding.	Yes	Mandatory
Transaction Date [String(16)]	Transaction Date & Time (YYYYMMDDHHmmss)	No	Mandatory
Channel [String (25)]	Value of “APICORPBC” to be passed	Yes	Mandatory
BcID [String (50)]	Client BCID. This will be provided by ICICI Bank during onboarding.	Yes	Mandatory

A Sample request packet for IMPS Name Inquiry	
Header	apikey = ‘----------' x-priority = ’0010’
Body	{ "BeneAccNo": "123456080", "BeneIFSC": "NPCI0000001", "TranRefNo": "20240221112300", "PaymentRef": "IMPSTransfer", "RemName": "Pratik Mundhe", "RemMobile": "9999988888", "RetailerCode": "rcode", "PassCode": "447c4524c9074b8c97e3a3c40ca7458d", "TransactionDate": "20240221112300", "Channel": "APICORPBC", "BcID": "IBCKer00055" }

Output parameters

Output parameter & type	Description
ActCode [String (4)]	ActCode Value
Response [String (255)]	ActCode Description
BankRRN [Number (12)]	It is a unique 12-digit number used to refer a particular transaction
BeneName [String (100)]	Beneficiary Name
TranRefNo [String(255)]	Transaction reference number

A Sample response packet for IMPS Name Inquiry	
Success	Failure
{ "ActCode": "0", "Response": "Transaction Successful", "BankRRN": "405211355307", "BeneName": "KAJAL", "TranRefNo": "20240221112300" }	{ "ActCode": "62", "Response": "Reason Unknown", "BankRRN": "406512367537", "BeneName": "", “TranRefNo”:202402211151558 }

UPI

Unified Payment Interface is an instant real-time payment system, which facilitates inter-bank, peer-to-peer, and person-to-merchant transactions

UPI Validation API:

The UPI Validation API is used for validating the Payee VPA and to know the bene bank IFSC code.

Input parameters

Header			
Input Parameter	Description	Provided by ICICI	Mandatory value
apikey	'API Key' is a unique value generated for each client. API Key will be provided to the client during onboarding and same needs to be used for all further transactions. Client must ensure that requests are being sent from whitelisted IP's only.	Yes	Mandatory
x-priority	This value states the priority order for the mode of payment. For UPI, x-priority = "1000"	Yes	Mandatory

Body			
Input parameter & type (maximum character)	Description	Provided by ICICI (Yes/No)	Mandatory value
Mobile [String(10)]	Mobile Number of the remitter. Mobile number provided during onboarding should be used.	No	Mandatory
Device-ID [String (50)]	This is a Unique device Token, which should be unique per channel user. This will be created during registration and it	Yes	Mandatory

	will be provided to the client by ICICI Bank.		
Seq-no [String (35)]	Unique random sequence number needs to be generated by client starting with ICI and without any special character. Merchant should keep the sequence number handy to check the transaction status.	No	Mandatory
Profile-id [String (10)]	Profile ID is created during onboarding and it will be shared with the client (remitter)	Yes	Mandatory
Channel-code [String (10)]	For composite pay merchants, value of channel code should be “MICICI”.	Yes	Mandatory
Virtual address [String (255)]	Virtual address of the payee to be passed	Yes	Mandatory
Payee-name [String (50)]	Beneficiary Name	No	Mandatory

A Sample request packet for UPI Validation	
Header	apikey = ‘----------' x-priority = ’1000’

Body	{
	"mobile": "7988000014",
	"device-id": "400438400438400438400438",
	"seq-no": "ICI59FC05C7438193",
	"profile-id": "2996304",
	"channel-code": "MICICI",
	"virtual-address": "pratikm1@icici",
	"payee-name": "PratikMundhe"
	}

Output parameters

Output parameters & type	Description
Success [String (5)]	This value of the output refers to the status of the transaction API call – success (true) /failure (false)
Response [String (4)]	Response code for the transaction
Message [String (99)]	Transactions status message – successful/failure (reason).
BankRRN [Number(12)]	It is a unique 12-digit number used to refer a particular transaction.
UpiTranlogId [String (10)]	Internal ID generated from the Switch.
UserProfile [String (09)]	Denotes profile ID of the user.
SeqNo [String (35)]	Seq-no of the transaction will be echoed back from the request.
MobileAppData [String (500)]	MobileAppData value will be Json based “STRING" (JSON – Stringified JSON.)
PayerRespCode [String (10)]	This is the response code populated by the Payer. For successful transactions or for transactions with no issue at the Payer side, this field will be blank
PayeeRespCode [String (10)]	This is the response code populated by the Payee. For successful transactions or for transactions with no issue at the Payee side, this field will be blank
PayeeRevRespCode [String (10)]	This is the response code populated by the Payee at the time of credit reversal. For all other cases, this will be blank.
PayerRevRespCode [String (10)]	This is the response code populated by the Payer at the time of debit reversal. For all other cases, this will be blank.

Sample response packet for UPI Validation	
Success	Failure

{
"success": true,
"response": "0",
"message": "Transaction Successful",
"BankRRN": "404749741416",
"UpiTranlogId": "349741416",
"UserProfile": "2996304",
"SeqNo": "ICI59FC05C7438193",
"MobileAppData": "SUCCESS,Mask Name=surya12347,IFSC=ICIC0000001",
"PayerRespCode": "",
"PayeeRespCode": "",
"PayerRevRespCode": "",
"PayeeRevRespCode": ""
}
	{
	"success" : true,
	"response" : "36",
	"message" : "Virtual address not present",
	"BankRRN" : "406551765858",
	"UpiTranlogId" : "351765858",
	"UserProfile" : "2996304",
	"SeqNo" : "ICI59FC0qss5C7438",
	"MobileAppData" : "",
	"PayerRespCode" : "",
	"PayeeRespCode" : "",
	"PayerRevRespCode" : "",
	"PayeeRevRespCode" : ""
	}

Encryption & decryption

For the purpose of maintaining security of APIs, it is very important for the APIs to be sent between the merchant and the bank in an encrypted form. Thus it is required for the merchant to initiate the API in an encrypted form using their public key. The bank will also be sending the response in the encrypted form and the merchants need to decrypt the same using their private key at their side and take the action accordingly.

Input parameters

Input value & type (maximum character)	Description	Provided by ICICI (Yes/No)	Mandatory value
RequestId [String (64)]	Unique identifier for the request. Not being stored at any level.	--	Optional
Service type [String]	Service Name; Identifying the backend service name	--	Mandatory
EncryptedKey [String. Base64- encoded data (case insensitive)]	One-time use AES key encrypted by the Client's public key. Requirement is for a 128-bit key (with 256-bit key supported as an option).	--	Mandatory
OaepHashingAlgorithm [String (6)]	Describes the algorithm used for Asymmetric Encryption of one time AES key.	Value: NONE, Meaning: RSA/ECB/PKCS1 is used for Asymmetric Encryption Value : SHA1, Meaning : RSA/NONE/OAEPWithSHA1AndMGF1Padding OR RSA/ECB/OAEPWithSHA1AndMGF1Padding	Mandatory

IV [String. Base64-encoded data (case insensitive) (24)]	The initialization vector used when encrypting data using the one-time use AES key.	Exactly 16 bytes actual value to match the block size	Yes, if IV is not part of encrypted data itself. Leave blank otherwise. For the response data encrypted at ICICI end, IV will always be part of encrypted Data itself and it is randomly generated for each request. Can be retrieved by Client by retrieving the first 16 bytes of Base64 decoded encryptedData
EncryptedData [String. Base64-encoded data (caseinsensitive)]	Contains the encrypted dataPayload object containing the business information. Encrypted by the ephemeral AES key using AES/CBC/PKCS5Padding. Sample unencrypted object: Please refer first section of document for a sample object.	--	Mandatory
ClientInfo [String]	ClientIP or other information	--	Optional
OptionalParam [String]	Reserved for future use	--	Optional

A Sample encrypted response packet sent by ICICI Bank to client (JSON):


```
{
   "requestId": "",
   "service": "",
   "encryptedKey": "pKY7abA+qRizRdYdAmZ8P+O0PIi+4NQSoZSEEEj/kamfyfxqGtleIn65SsMJrLSPI
XtHJrjrMYoMphUXoV0zwFMl7v7AMiGgs7y2LntrDxMdOyNk7XTENouO0i7sEHmgyZWqS/y0sQQhnk
1rWEc9BZaiVBpNPsjccAZ9nx4GyRyLvso0pdMrpbpdj8zO80l+5pMAnyf1WcGx5DbZv6fRnl/OddklQ
eAQASsSG0oFoQcW8pGszg3oDa6OA0Kbd+7om4CvbcueArU6+MbH3aY3af5isGd0w7/BPTmq8Fei
2TbHpSFLRl37nqprRL6+JWk+4eKWd0xVixxFnNtli4FDh+mnkC6CHBriC4O1hnoSpVken6wl+ltlKNa/
FFOE8RucfmBdrHF99eJASuNnU+k8fbcQKPo7gkAtavAAaN/khaQ/Msa6VEOEWM2lndUpX1PH+dZ1
pNFunRWrn813NfodIAtZ1V6KfWT/Rj1DvCsSD/aJgSr3nf1Hz9WUl9JvAtMeWBwMH+0OaRnrtoJC8
Sp76Rfxm/Wb5ddQXavE8Oe4AfvvDK3IZ1hwdQYxa3NFuE9/nTXSOKsmuNrcllhnsDNmp6ndjXwr/jl
KwH9Fw9i3LYVJL80ng55p9F1VboijVjqhm9FmSIKtmKVgXIL04lytbsm8Z1BHFU3ajvkrUNsYJrg=",
   "oaepHashingAlgorithm": "NONE",
   "iv": "",
   "encryptedData": "mtlktFRBPPs6SVxh0RWWvsa9bj/W/M56QwvbY/vgYAIolBZuL8L0Qkq8Rxo24
U47Tq77UXDWX+F1KCpWddxGMwnvqCL3k+2b4jJNKsWSVWykXuF3pVRGH4bKZ9gKfRh9C7wQ
9x6SuguXUDhjFp+Dss7przaNBY0GsRKXR+vUlvR1pwDW0Np+tYvkX7C0n17wg7GGbuWfN5nf92a
9xjj512ikE3615pUyDVHBNKeUNmRHue/UB6lY3nB/+/JC/Wro",
   "clientInfo": "",
   "optionalParam": ""
}
```


2.1 ENCRYPTION-DECRYPTION PROCESS

2.2 For encryption of request at ICICI:

SesionKey = Randomly generated string of length 16 (OR 32).

encryptedKey = Base64Encode (RSA/ECB/PKCS1Encryption (SesionKey,ICICIPubKey.cer))

IV = Initialization Vector- Exactly 16 bytes actual value to match the block size

encryptedData = Base64Encode(AES/CBC/PKCS5Padding(Request,SessionKey, IV))

2.3 For encryption of response at ICICI:

encryptedKey= Base64Encode(RSA/ECB/PKCS1Encryption(SesionKey,ClientPubKey.cer))

Session key is nothing but randomly generated string of length 16 (OR 32).

encryptedData = Base64Encode(AES/CBC/PKCS5Padding(Response,SessionKey))

2.4 For decryption of response at Client:

IV=getFirst16Bytes(Base64Decode(encryptedData)

SessionKey= Base64Decode(RSA/ECB/PKCS1Decryption(encryptedKey,ClientPrivateKey.p12,))

Session key is nothing but randomly generated string of length 16 (OR 32) .

Response = Base64Decode (AES/CBC/PKCS5Padding Decryption(encryptedData,SessionKey,IV))

OR

2.5 Steps for Encryption

Generate 16-digit random number. Say RANDOMNO.

Encrypt RANDOMNO using RSA/ECB/PKCS1Padding and encode using Base64. Say encryptedKey.

Perform AES/CBC/PKCS5Padding encryption on request payload using RANDOMNO as key and ivinitialization vector. Say encryptedData.

Now client may choose to send IV in request from one of below two options.

a.) Send Base64 Encoded IV in "iv" tag. (Recommended Approach)

b.) Send IV as a part of encryptedData itself.

bytes[] iv = IV;

bytes[] cipherText = symmetrically encrypted Bytes (step3)

bytes[] concatB = iv + cipherText;

encryptedData = B64Encode(concatB);

Perform AES/CBC/PKCS5Padding encryption on DATA using RANDOMNO as key and Base64encoded RANDOMNO as IV. Say ENCR_DATA.

2.6 Steps for Decryption

Get the IV- Base64 decode the encryptedData and get first 16 bytes and rest is encrypted response.

bytes[] IV= getFirst16Bytes(Base64Decode(encryptedData)

Decrypt encryptedKey using algo (RSA/ECB/PKCS1Padding) and Client's private key.

Decrypt the response using algo (AES/CBC/PKCS5Padding).

Ignore first 16 bytes of response, as it contains IV.

APPENDIX

Changes incorporated in Composite Validation Document v1.01

1. Introduction

- Introductory text of Composite Validation API
- Endpoint API 'LIVE' URL

2. API Details

2.1 Input Parameters:

- Addition of Input Parameter 'Type' and 'Maximum Character' for all Validation API
- Addition of 'Provided by ICICI Bank' column for all Validation API
- Addition of 'Mandatory Value' column for all Validation API
- Addition of Input Parameter 'APIKEY' for all Validation API
- Defined the 'Description' column for all Validation API

IMPS Name Match API:

- Removal of the parameter 'Sender Name' from IMPS Name Match API request packet section as it is a redundant field and is not mandatory.
- A value of 'APICORPBC' to be passed for channel

IMPS Name Inquiry API:

- A value of 'APICORPBC' to be passed for channel

2.2 Output Parameters:

- Addition of 'Type' and 'Maximum Character' for all Output Parameters of all Validation API
- Changes of 'Description' column for all Validation API.

UPI Validation API

New Parameters Introduced in Response Packet

- 'PayerRespCode'
- 'PayeeRespCode'
- 'PayeeRevRespCode'
- 'PayerRevRespCode'