Aerion PG Rupay Service

API Document

For Merchants / Payment Aggregators

Version 1.0

Last Updated:

20th Sep 2023

Revision History

Version Number	**Modification Made**	**Modified By**	**Reviewed By**	**Date**
1.0	Initial Draft	Vinod	Ramesh	25th Oct 2022
1.1	Added ALT ID Changes	Amritha	Ramesh	20th Sep 2023

Table of Contents

1 Introduction

enStage has created an internet payment – ACCOSA-PG that offers payment services to Billers, Merchants and Consumers.
This document describes how a merchant will have to integrate with enStage's ACCOSA-PG – Rupay API Service.

For Rupay PG transactions merchant has to integrate five API

- BEPG Checkbin API
- Initiate2 API (Redirection Flow)
- Generate OTP/Resend OTP (Seamless Flow)
- Verify OTP
- Authorization (BEPG Authorize) API

This document explains about the authentication and Generate/Resend OTP processes.

2 Transaction Flow for Rupay Transaction

- Wibmo calls Rupay BEPG CheckBin API of Accossa PG with certain details mentioned below in request format for Rupay Transaction.
- The checkbin & Authorization request/responce format remains same for both Redirection and seamless flow.
- Based on the response received as part of BEPG_checkbin request , merchant calls Initiate2 / Generate OTP API call to Accosa PG.
- After receiving request from the merchant/wibmo, Accossa-PG communicates with NPCI and gets details of authentication (RedirectionURL using Initiate2 request /OTP using Generate OTP api) and responds back to the merchant.
- ACCOSA-PG disassembles the transaction information to confirm transaction origination and stores details of the transaction into its databases
- In case of redirection flow, merchant redirects to the ACS URL received as part of Initiate2 response from AccosaPG along with all the required parameters. Accosa PG will share all required parameters to redirect to the ACS URL as part of Initiate2 response.
- AccosaPG will validate the response received from redirection and send back the response to merchant . Merchant can initiate the authorization request post successful redirection response.
- Verify OTP request along with request ID and otp is initiated by merchant which is in turn sent back to the NPCI by ACCOSA-PG and based on the otp verification response , if failure – is intimated back to the merchant by PG using status values and description as per Response format for Rupay Transaction, if Success – the same is intimated back to merchant
- Authorize request is initiated by merchant to PG to NPCI to finally send back the authorize response to merchant.
- Transaction Status Code of 50020 and pg_error_code of 0 means authentication of this transaction was successful and any other status code or error code means it failed. Refer to appendix for more details

3 API Gateway

All the requests must go through API – gateway which accepts message level encryption of the payload.

POST: https://pgdomain/api/v1/{serviceName}

Request Header

Parameter	M/O/C	Sample
x-api-key	M	Actual api-key will be provided from PG during Onboarding - 777a9184-2e70-11ed-b928-005056b59d84
pgInstanceId	M	Acquirer Id provided by Wibmo – Example 8645
merchantId	M	Same as pgInstanceId

Request Body

Parameter	M/O/C	Sample
signedEncRequestPayload	M	Request PayLoad Signed with acquirer PrivateKey, encrypted with AES256 KEY
requestSymmetricEncKey	M	Encrypted AES256 KEY with Wibmo's Public Key
iv	M	Initialization Vector (IV) value
pgInstanceId	M	Acquirer Id provided by Wibmo – Example 8645
merchantId	M	Same as pgInstanceId
clientReferenceId	C	Reference id generated at client end. If not send our API will generate it.
var1	O	
var2	O	
var3	O	

Sample Request

```
{
  "signedEncRequestPayload" :
"xFsQAMFBmTmA1+7X3b8u+Vyul/z5yr3iz3aISnWjfNjGo3AzES9PAj7its52NZCsrYaYkxcH2b+SPm1Jt6LFgyci07wy7UzcoNQu4KN9L0giYtHa8sAF5rIN5
xUqOiRw7yydf20TARf63Y5dMLCYeT/G+2NCgwUWKNVGgMhicm2pK+uq12IrPITKt3Qrm7ksbhxqI76N4NAGw4SUT6bAezMfPgwcBcsU922soKA8/zf3Z/lj1lk
lZnsaeONs08iAv7yZlp1fguOZR/RIF//JuGFkYl5fWvO6/LWiOmFokAckWW3GfqGW47B+lXXXMCcX0O/H0NTmywshe6ZsP1PXAY1zpakxIhKuAVJM7SlvROHGd
b4x6YU07p/rBaj/ksY5tGwL/HeHOfY1pNYcAOdN3pU5c5QLwBMUM2+esY6XfyMXpmqU6CqRVWaeU9Rybv+1+Uq1Rq4heVYbEnAXT7t1ih7zGRlG6CnE/SxPgNa
UAYBSD/2ZjykD57hWUabbhkDHF8ELLTphzOiVmym1QN3zESbwVrHFjjZkGT3aelexxIEVndeI5b7ZaYf2JPo/SMfaCibeAYOGadUNwJNd2h2qwAe4+3HyT6FXo
Dibyl7yKGZKx5yjiQWL06Wto3GI+Itl9aZ1bOGk6i7pxjbbV/D4x0LI6BblWSp4pr/4df8aUuTSZWqn6gN8BIEx8wbFWjbDsHyPFAz3+WRAEJ8eZzpgXn/Ik4C
n6/42mkUJYRiCr+bAqUgpMizGFmKWG9/UX1gxcAXvAGhGDzBhDn3iDg0dz4wnvkQKCCaYnt828iE50Vepq2cU6PdD5403ki5WmL0l9ToblErbOHiWQy7+PZ4ho
GKR5eQ3hhkfiwbXlqidYxH1WfJrUijVH9lH06d+YLu44YWj0WlRjZ5mP3d+oTs2KDujRqGab2NfD643fKYRhkU+atyJ9aKtlO9OB370XoL+d1g4GkFStC7y8P6
iJSZ6Jp9Jew7sTbOzTsLL/OQ2ZuzBsGvn4WlAp3vsj1mh8GdEjG2QQPgYxfMa+4z3FFmbWznLmIhuIVvQmlvEWGeInQb9pzN+Atnt4P/gHQ9sbEDwR4Yw6CcXq
qBmdlX0vIM9/7wXHksWLPpbx+80VwMRV8XYUh40kPJZlXmbUv981mu04h/mFIysXJUyXyRc3Ah09RPfKMqfTYwYNuVJ43YTjgqS01UyZj4ntcwO86t+1S39NoW
dtssDUQO9TFqvPdqRYzrUE7nqWaBQnmknN906fCFBJWx8PMKCdpdmLvUM39surndFpFkwqowibvyjhqi6aa+C5D5U0rOZtFyLTFKAzaHyunm1iqjs2LqtQcL5M
L+SIcSroPU4Bp70AfM7zgHu1YXRc9jfJFFfOadPSyClkzbs8pz3VAHxZVV+1Tm+Ry7gNjePkLJJSuIC8nNh2inTzpgCmNYHaUBQxko2tomtwWBO3SEn1ZfvSWL
ud1Wx7D8OgtXvPhVNz6VAfJlrTuB7Zhz6L45CoX0oJb5IlVpdwyXbQLlbMrp/N4kKT2EpPuIwkMFfpsvd6sDFJKnlGYqpFRtvnTcRJM/dlq5Y0+mm+yEunryZl
XK07AZal935MwPDG7g5FGVQ0HPlZ+o131xpbihGz4h6NLRbnf25uo9EFFHk9q4DrRKqSmk9q/nQNko8eygu+nN2RtAeydqLiRSaLUG3iimpJWkWij4/d3ez524
QkKBPVRysnLuEMHD/GfnNyGx+XOneHDObzISly+8kFt6gWugucEmPnT1h2C71/iXbFF7dcu5qi1+0BHfIDNzac8vXOpvi82glNi6HLqBtYn30ZRrctGzQrh9U8
U5JLPaEai6TS1jILmrdIDJlW0oQl0PUQm2q1QoHW16ZT2vwwhyXCD7+2wHhcm4mdr7zh2/q0LENNtW4khjJ13aZPaqprkk66pLO3JxvMMyrs7wzspYqvuxJ7lT
2zGcPptnNioPY8Gs9Mn32mZB8UT439PKnbL5C4f5hEUPsBp9htXLWX+ZEELFoyoISU+tlPrqo//H6Lr3sXaVeQSsmCYMS2W6hFi43Loe5coWA0P8ClUujzSL6H
0mfF1p9QkaTy6q5zzvwKEO9X4z8+r+evRIZ6nNcE8J+12map6vxuw==",
  "requestSymmetricEncKey" :
"cHOj205sox0eAFUEajWravTUzAP+5b1Yk7rbyIvxszj64hbFHQjxVS8MATGgioC0i8kso6aDPgckesWZCMAtOERWbYeEY3nA8AXr/8pRlZntiowbqqUTHodj1
hTh+7+wOJ3Eo614RVgIEru/AZ0fVq5c9oIL+8QmGj9gFJpO4GRAr++lx2eee2LcsXoZk6NcbrevJCogjxvVFrBrq0sDpriI3KMCo8bgTiG7TMyB+TSc/dx6En4
JEDhy5iqws4lJcOw8C/JZDu9qb8NqG1jalPb7cGrHarOrfNMWOEwu4jFeUEONb37Y6CnwT39sykcgOLSFqSTH4O3CDzbuDLnsBA==",
  "iv" : "nbXTfdxxouQoOllf",
  "clientReferenceId" : "aaddvkkllllle123344”
}
```

Response Message Body

Parameter	M/O/C	Sample
signedEncResponsePayload	M	Response PayLoad Signed with wibmo PrivateKey, encrypted with AES256 KEY
responseSymmetricEncKey	M	Encrypted AES256 KEY with Acquirer Public Key
iv	M	Action To Be Performed
clientReferenceId	M	Same Request clientReferenceId will be returned
statusCode	M	Transaction status is success or failure
statusCode	M	Transaction status is success or failure
var1	O	
var2	O	
var3	O	

Sample Response

```
{
"signedEncResponsePayload":
"LTPpSHNwAuEthTTDlmO0G9QY5dKfT7COTFaJe+7FhKkrgVYrSkJIxQeFWhi0tEghUDm54yE4QokEk42ifRpenCcGC1fLF+ReOsErEhCEHy1mUUE+3
4soWWdhXqEY3GflWyJK6v7sr1A6UVKgJCraSpJy5Ubhzrxtt10Jh/bojo6NJysRQszgEnYUagzDqW5YVIjgG9InJ/2meJpQt3aCx9scgHG6A8iJBq N
maJYg1cjB4KVRZGFzRSu3TKF3miSmwG7CU1qC8+9F+DyD2ExQV9/8puN5bpVlJkiOnye1GGzyBh3LXXGYDS7yaKoR6joGsPuFrJN48uZy9UqrS/qva
Co9/zGhh2gZEe1qxGNxV649AW5pdQMdrvRGgYRghioAOtgLJneSnADjO5L4xWEn1wnY/QjTxpCPh52n4K7PxtjxPS6wOTZaqOUY32+fn7Fn3O3pjBr
5go6Dasynkl2UeRs4obc5ReUD3AwSonvqVAtLCyApmmCUAHZloCWVd20NHubzQRPJHPmHjx+bmTn6etPO2w/rS4Y1KhGlSQ5WzDUJzYh3MMOhV7aKX
TeKmbxLuEKQebZl8ajEIv3DKLZjAk5cmuFk8el8TuiTpsXTxnqR7Azapc2JZHZBBoMvhcpW/1ACnbsdI3vc9kltEQThRWXqhuyoIMdijUifUmNAScS
tbZMMPGbNdx1tey313YiP6awakCCHyEyFHsmofOPQRMxxoMe
5z32UF1Ky/mzVOde1Abn8ZLJuFw/NI7fSYLfYTKQxklGe+vjmB0czti96TuUc0A9rSFhvXJUjjwQpiuzLbT9fmQjzh+WLl7lVSf7Tc21bePN4ZYQv9
jF24SVeN8jsMrtUXgkYShucUmNDcpiBzCyWYjKH/HG1MMGhhEKpBo7xScrdqZ8k7/8jf1mXBqwe+CwHBjewCV55GUaTBJBhorv5ftIqQCNvvDXDvWe
Ea/OJ2f/Ze8oNzi1KcRlKkFhIqzQAuUUCAQIJ3wnIwfBlpRXXqbj4mK2M0/CUoMUKpARe+VLJFmF7kPmp36gaY2oi2cFwmtshPGX/JZGR9va5u2Rps
329AoA1DMyvsh0SoOcZp54X8qYKg8EXlG+TLVlqcOc4fN4oSwMkV8JvT1s0nu7/bSNR6afRI12omVxYml3VljKoGNRCgMN2R+LYi5ziOvFm9kO4IdF
1vjQhAqxH8TvuqCX2zgqjNwd/1Nbwfx2wl/ufMd/BsjkbUOQKt0CZTt8y7vYGh3gCMCpDxIG1dirLED+liVFaZybzhZhQUHG3otiDNhDfjQJn/
3B0CKG6YCdHJjUg",
    "symmetricEncKey":
"iPeXe2KXche23UZHURwFw+BdORL5xSPp66wXVJfAQRNcKR+nnwZ4dOsAKJX3dmSQ0lx6Cp6MpZBZo2UQH2eQJbt1SJmFK2pPFNXD+L7wYcBs/p4xg
4xk8l7HK7E1qkxoJEsKS9DunqYvl4IjW/GVS9G2+rHSEB2VrRlP4Q3k++Tb8Rp94nFjSv6wYIz4ghZdLllqnbw056rAG9mOF3i00box5kGZdX4xG+o
LjT/GSzi56mx2SvctNXmiB4N9+6BC7GxbkCc4lLYhRa9MDTPcoz1HkF1Cjcp+ldYGZsYf3KllfmmrVeIlBq8F9ABUzwZjot3GE12gX4EdZSnR
ryabfA==",
    "iv":"Ob06nKjivp+plDCY",
    "clientReferenceId": "aaddvkkllllle123344”,
        "statusCode": "PG99200",
     "errorMessage": null,
     “var1”: null,
     “var2”: null,
     “var3”: null,
}
```

Error Codes	Error Message	Http Status Code
PG99401	Attempted login from unauthorized IP	401 Unauthorized
PG99405	Request Method Not Allowed for API Access	405 Method Not Allowed
PG99429	DDOS Attack Detected from Requestor	429 Too Many Requests
PG99410	Cross Site Forgery Detected in API Request Message Payload	410 Gone
PG99400	Code Injection Detected in API Request Message Payload	400 Bad Request
PG99503	API Back-end Service Not Available or Timed-Out	503 Service Unavailable
PG99429	API Access Quota Exceeded	429 Too Many Requests
PG99500	Backend Service Provided Unexpected Response	500 Internal Server Error
PG99401	Invalid Client Certificate	401 Unauthorized

PG99401	Scope Validation Failed	401 Unauthorized
PG99401	Invalid API Key	401 Unauthorized
PG99401	Signature verification Failed	401 Unauthorized
PG99400	Decryption Failed	400 Bad Request
PG99400	Bad Request	400 Bad Request

4 Request Format: Rupay Checkbin

*M/O/C = Mandatory/Optional/Conditional

Field Name	M/O/C	Type & Size	Format	Description	Example
pgInstanceId	M	N(10)	Variable	Unique Id of PG	16787434
cardBin	M	N(9)	Fixed	First 9 digits of a valid card. The Bank Identification Number, BIN, of the card the consumer entered at time of checkout on the merchant site.	XXX652851000XXX

5 Response Format: Rupay CheckBin

*M/O/C = Mandatory/Optional/Conditional

Field Name	M/O/C	Type& Size	Format	Description	Example
status	M	N(5)	Variable	Transaction Status	success failed
errorcode	M	N(1-4)	Variable	Error Code	0 = No Error (any other value means some kind of error, full error codes list shall be provided in appendix B,C)
qualifiedInternetpin	M	AN(4 or 5)	Variable	Identifies the BIN eligibility for RuPay eCommerce transactions. True represents BIN is eligible.	True / False

authenticationNotRequired	M	AN(4 or 5)	Variable	If BIN is eligible for eCommerce transaction and if this field is True, then acquirer can call the bepg_Authorize API call directly without the need for authentication. This will be set to true for international card brands supported on RuPay rails.	True/False
availableAuthMode	C	AN(2-30)	Variable	Will not be available if authenticationNoRequired is true. Identifies the available Authentication mode for this BIN. If the BIN supports multiple authentication modes, then the same will be sent as a comma separated values.	Comma separated values of authentication modes. e.g., 01 (OR) 02 (OR) 01,02 Where 01 – Redirect 02 – S2SOTP
additionalProductsSupported	C	AN(2-30)	Variable	Will not be available if authenticationNotRequired is true. Identifies the additional products supported by the issuer for the card BIN. If the BIN supports multiple products, then the same will be sent as a comma separated values.	Comma separated values of products supported. e.g., 01 (OR) 02 (OR) 03 (OR) 01,02 01,03,04 Where 01 – EMI 02 – SI 03 – SFA 04 - MOTO
errormsg	C	AN(0-(1024)	Variable	Description for the error code.	

6 Redirection Flow:

A. Initiate2 Request Format:

*M/O/C = Mandatory/Optional/Conditional

Field Name	M /O /C	Type & Size	Format	Description	Example
pgInstanceId	M	N(10)	Variable	Unique Id of PG	16787434
merchantId	M	N(8)	Variable	PG Merchant ID	44065131
merchantReferenceNo	M	AN(1 -20)	Variable	Unique Reference number of a transaction, Over 50 in length will be truncated	AX143565
pan	M	N(13 – 19)	Variable	Should be a valid Rupay card, all the digits as a continuous string without formatting.	XXX6073849800604961XXX
cardExpDate	M	N(6)	Fixed	The month and year the card expires as embossed on the card Format: MMYYYY	XXX122024XXX
browserUserAgent	M	ANS(512)	Variable	Acquirer will collect the UserAgent string information of the cardholder browser/app. i.e, HTTP Header - User-Agent	User agent from browser

ipAddress	M	ANS(50)	Variable	Public IP address (IPv4 or IPv6) of the cardholder, end user who has performed transaction)	Eg: 103.14.160.193
httpAccept	M	ANS (256)	Variable	Accept header is used by HTTP clients (browser/app) to tell the server what content types they will accept and it should capture from end user device who has performed transaction. i.e., HTTP Header : Accept	Eg : text/html, applicatio n/xhtml+x ml,applica tion/xml;q =0.9,image /webp,im age/apng, */*;q=0.8
authAmount	M	N (1-12)	Variable	Purchase amount including the minor units of currency with all punctuation removed i.e., implied decimals and no formatting For Token transaction if purpose ofAuthentication is "only Card registration" then amount should be passed is 0(zero)	e.g., 10025 OR 0 (For Token transaction if purpose ofAuthentication is "only Card registration")
currencyCode	M	N(3)	Fixed	ISO standard 4217, currency code in which the auth_amount is expressed.	356
cvd2	M	N(3-4)	Variable	consisting of 3 digits and is located on the back of the card within the signature area. Card Verification Data. (i.e. "0310")	Original Value XXX123XXX

orderDesc	O	ANS(1-50)	Variable	Order Description	order2JE8DCB
ext1	O	AN(1-16)	Variable	CTX Id, which is unique for every Transaction	123456
ext2	O	AN(1-16)	Variable	MTX Id	234565
amountInInr	O	N(1-12)	Variable	Used to Capture the base currency amount i.e. INR for International transactions in implied decimals	20000
originalAmount	O	N(1-12)	Variable	Mandatory for Convenience Merchant.Original sale amount is captured in this field in implied decimals.	20000
merchantResponseUrl	M	AN(500)	Variable	Fully Qualified URL to send back the response of Redirection request to NPCI. Merchant needs to form this URL where the response of redirection request will be sent back.	Fully qualified Merchant URL?Name1=Value &Name2=Value &Name3=Value
transaction_type_indicator	M	A(3)	Fixed	SMS- SMS Authorization Message DMS- DMS Authorization message AUT- Authentication only SGC - SMS authorization & with guest check out DGC - DMS Authorization& with Guest checkout	SMS (or) DMS (or) AUT (or) SGC (or) DGC

purposeOfAuthentication	C	AN(1-38)	Variable	purpose of the Authentication used so that PG can send required data in Initiate or generate API. This field is Mandatory for ALT ID and Token Transaction. This is case sensitive	Pass either of the values 1. Only Card registration (OR) 2. Card registration during Authorization (OR) 3. Real Card Transaction (OR) 4. Token Transaction (OR) 5. ALT ID Transaction
tokenAuthenticationValue	C	AN	Variable	To be passed by merchant for token PAN transaction to be sent to Card union	

B. Initiate2 Response Format:

*M/O/C = Mandatory/Optional/Conditional

Field Name	M /O /C	Type & Size	Format	Description	Example
pgTransactionId	M	N(1-8)	Variable	PG Sale Id	866
rupayTransactionId	M	N(30)	Variable	Rupay Transaction Id	102345678910234567891023456789
redirectURL	M	ANS(1024)	Variable	The URL the cardholder needs to be redirected to initiate authentication process and parameter names. As part of this field AccuGUID, AccucardholderId, AccuHkey will be passed along with Issuer URL to redirect Note : Values are html encoded.	Fully qualified URL?Name1=Value &Name2=Value &Name3=Value

accuRequestId	M	ANS(1024)	Variable	To avoid tampering of request data in transit, Wibmo will provide a hash code for the issuer to validate. This value needs to be passed along with other required parameters in redirection request.	
session	M	AN(500)	Variable	To handle session at wibmo end, wibmo can populate any value and same value should be passed as one of the parameters in redirection Request.	processRupayAPI|866|AX143565|58543624|767348378
status	M	AN(7)	Fixed	Transaction Status	success/ failed
errorcode	M	N(1-4)	Variable	Error Code	0 = No Error (any other value means some kind of error, full error codes list shall be provided in appendix B,C)
errormsg	M	AN(0-(1024)	Variable	Description for the error code.	

Redirection Request Parameters: (To call ACS URL with all required parameters)

Based on Initiate2 API Response, merchant needs to construct redirection request along with required parameters and post to the RedirectURL received

URL: The RedirectURL (excluding the parameters) received in initiate2 response
Method: POST
Parameter Name & Type: Hidden

Parameter	Description
AccuCardholderId	Generated by Issuer/NPCI BEPG and extracted from Initiate2 API Response in "RedirectURL" tag
AccuGuid	Generated by NPCI BEPG and extracted from Initiate2 API Response in "RedirectURL" tag.
AccuReturnURL	Fully qualified Acquirer return URL that issuer will use to redirect the cardholder back, upon completion of the authentication.
session	To handle session at wibmo end, wibmo can populate any value and same value should be passed as one of the parameters in redirection Request.
AccuRequestId	To avoid tampering of request data in transit, the Wibmo will provide a hash code for the issuer to validate. This hash code can be generated by hashing Transaction ID, AccuCardholderId, AccuGuid, and session parameter value.

Note: Parameter Names in redirection request and response are case sensitive.

Redirection Response : (From Wibmo to merchant)

Field Name	M/ O/ C	Type&Size	Format	Description	Example
pgTransactionId	M	N(1-8)	Variable	PG Sale Id	860
status	M	A(5)	Variable	Transaction Status	success failed
errorcode	M	N(1-4)	Variable	Error Code	0 = No Error(any other value means some kind of error, full error codes list shall be provided in appendix B,C)
errormsg	C	AN(0-(1024)	Variable	Description for the error code.	

7 Seamless Flow:

A. Request Format for Generate OTP

*M/O/C = Mandatory/Optional/Conditional

Field Name	M/O /C	Type& Size	Format	Description	Example
pgInstanceId	M	N(10)	Variable	Unique Id of PG	16787434
merchantId	M	N(8)	Variable	PG Merchant ID	44065131
merchantReferenceNo	M	AN(1-20)	Variable	Merchant Reference Number	AX143565
pan	M	N(13-19)	Variable	The full length card number, (PAN), as entered on the Merchant website by the cardholder.	Clear Card Number 5213123402341234XXX
cardExpDate	M	N(6)	Variable	The month and year the card expires as entered on the merchant website by the cardholder.	MMYYYY 092023 XXX
cvd2	M	N(3-4)	Variable	The CVD2 as entered by the cardholder on the merchant's website. This field will not be received for an InApp transaction. The same is mandatory in all other cases	CVD2 as displayed on card 123XXX
nameOnCard	M	AN(1-50)		Name of a card holder	Vinod

cardHolderStatus	M	AN(2)	Fixed	NW – Transaction performed at domestic merchant. IT – Transaction performed at International merchant. AT- Authentication Transaction Only DI –Domestic InApp Transaction Only DI –Domestic InApp transaction II – International InApp Transaction GC – Domestic Guest Checkout Transaction	NW, IT, AT, DI, II, GC
amount	M	ANS(1 - 12)	Variable	Purchase amount including the minor units of currency with all punctuation removed i.e., implied decimals and no formatting For Token transaction if purpose ofAuthentication is “only Card registration” then amount should be passed is 0(zero)	e.g., 10025 OR 0 (For Token transaction if purpose ofAuthentication is “only Card registration”)
currencyCode	M	N(3)	Fixed	ISO standard 4217, currency code in which the amount is expressed.	356 only – potential for additional in future
customerIpAddress	M	ANS(7-39)	Variable	Shopper browser IP address	203.11.11.12
browserUserAgent	M	ANS(512)	Variable	User agent from browser	Acquirer will collect the UserAgent string information of the cardholder browser/app. i.e, HTTP Header - User-Agent
httpAccept	M	ANS (256)	Variable	Accept header is used by HTTP clients (browser/app) to tell the server what content types they will accept and it should capture from end user device who	Eg : text/html, 72pplicationn/xhtml+xml, application/xml;q=0.9, image/webp, image/apng, */*;q=0.8

				has performed transaction. i.e., HTTP Header : Accept	
ext1	O	AN(1-16)	Variable	CTX Id, which is unique for every Transaction	123456
ext2	O	AN(1-16)	Variable	MTX Id	234565
amountInInr	O	N(1-12)	Variable	Used to Capture the base currency amount i.e. INR for International transactions in implied decimals	20000
originalAmount	O	N(1-12)	Variable	Mandatory for Convenience Merchant.Original sale amount is captured in this field in implied decimals.	20000
purposeOfAuthentication	M	AN(1-38)	Variable	purpose of the Authentication used so that PG can send required data in Iniate2 API and generate_otp call. This field is Mandatory for ALT ID and Token Transactions.	Pass either of the values 1. Only Card registration (OR) 2. Card registration during Authorization (OR) 3. Real Card Transaction (OR) 4. Token Transaction (OR) 5. ALT ID Transaction
tokenAuthenticationValue	C	AN	Variable	To be passed by merchant for token PAN transaction to be sent to Card union	axqwioyuricvvXw==

B. Response Format for Generate OTP

*M/O/C = Mandatory/Optional/Conditional

Field Name	M/ O/ C	Type & Size	Format	Description	Example
status	M	N(5)	Fixed	success/ failed	Transaction Status
validityPeriod	C	N(2)	Fixed	The validity of the OTP in minutes for this transaction. Any newly generated OTP for the same transaction will automatically invalidate the old OTPs.	00 to 30
errorcode	M	N(1-4)	Variable	0 = No Error(any other value means some kind of error, full error codeslist shall be provided in appendix B)	This is a numeric code used to provide success or failure response
errormsg	C	AN(0-(1024)	Variable	Description for the error code.	
pgTransactionId	C	N(1-8)	Variable	Sale Id	860

8 Request Format for Resend OTP API:

*M/O/C = Mandatory/Optional/Conditional

Field Name	M/O/C	Type&Size	Format	Description	Example
pgInstanceId	M	N(10)	Variable	Unique Id of PG	16787434
merchantId	M	N(8)	Variable	PG Merchant ID	44065131
pgTransactionId	M	N(1-8)	Variable	PG Sale Id	860
merchantReferenceNo	M	AN(1-20)	Variable	Merchant Reference Number	AX143565
cardHolderStatus	M	AN(2)	Fixed	NW – Transaction performed at domestic merchant. IT – Transaction performed at International merchant. AT- Authentication Transaction Only DI –Domestic InApp Transaction Only DI –Domestic InApp transaction II – International InApp transaction	NW

9. Response Format for Resend OTP

*M/O/C = Mandatory/Optional/Conditional

Field Name	M/O/C	Type& Size	Format	Description	Example
status	M	N(5)	Fixed	success / failed	Transaction Status
validityPeriod	C	N(2)	Fixed	The validity of the OTP in minutes for this transaction. Any newly generated OTP for the same transaction will automatically invalidate the old OTPs.	00 to 30
errorcode	M	N(1-4)	Variable	0 = No Error(any other value means some kind of error, full error codeslist shall be provided in appendix B)	This is a numeric code used to provide success or failure response
errormsg	C	AN(0-(1024)	Variable	Description for the error code.	
pgTransactionId	C	N(1-8)	Variable	Sale Id	860

10 Request Format for Verify OTP:

*M/O/C = Mandatory/Optional/Conditional

Field Name	M/O/C	Type&Size	Format	Description	Example
pgInstanceId	M	N(10)	Variable	Unique Id of PG	16787434
merchantId	M	N(8)	Variable	PG Merchant ID	44065131
merchantReferenceNo	M	AN(1-20)	Variable	Merchant Reference Number	AX143565
pgTransactionId	M	N(1-8)	Variable	PG Sale Id	860
otp	M	N(4-10)	Variable	The otp entered by the card holder.	4 to 10 digit OTP

11 Response Format for Verify OTP:

*M/O/C = Mandatory/Optional/Conditional

Field Name	M/O/C	Type&Size	Format	Description	Example
pgTransactionId	M	N(1-8)	Variable	PG Sale Id	860
status	M	A(5)	Variable	Transaction Status	success failed
errorcode	M	N(1-4)	Variable	Error Code	0 = No Error(any other value means some kind of error, full error codeslist shall be provided in appendix B,C)
errormsg	C	AN(0-(1024)	Variable	Description for the error code.	
tranCtxId	C	N(30)	Fixed	Transaction context id . A uniqueID assigned by NPCI.	1233456789012..
authResponseCode	C	AN(10)	Variable	Authentication status code generated by issuer	

12 Request Format for Authorize API:

*M/O/C = Mandatory/Optional/Conditional

Field Name	M/O/C	Type&Size	Format	Description	Example
pgInstanceId	M	N(10)	Variable	Unique Id of PG	16787434
merchantId	M	N(8)	Variable	PG Merchant ID	44065131
pgTransactionId	M	N(1-8)	Variable	PG Sale Id	860
merchantReferenceNo	M	AN(1-20)	Variable	Merchant Reference Number	AX143565
altId	C	AN(1-20)	Variable	Alt ID Pan Number, Mandatory if purposeOfAuthentication was sent as "ALT Id Transaction" in initiate or generateOtp Call	xx6123111142351711xx
altExpiry	C	AN(1-20)	Variable	Alt ID Expiry, Mandatory if purposeOfAuthentication was sent as "ALT Id Transaction" in initiate or generateOtp Call	122024
tokenAuthenticationValue	C	AN(1-20)	Variable	Alt Id Tavv, Mandatory if purposeOfAuthentication was sent as "ALT Id Transaction" in initiate or generateOtp Call	axEqregg==

13 Response Format for Authorize API:

*M/O/C = Mandatory/Optional/Conditional

Field Name	M/O/C	Type&Size	Format	Description	Example
transactionId	M	N(1-8)	Variable	PG Sale Id	860
status	M	N(5)	Variable	Transaction Status	50020 = success (any other value means some kind of error, full status codes list is provided in appendix A)
pgErrorCode	M	N(1-4)	Variable	Error Code	0 = No Error(any other value means some kind of error, full error codeslist shall be provided in appendix B,C)
pgErrorDetail	C	AN(0-(1024)	Variable	Description for the error code.	
approvalCode	C	N(6)	Variable	Approval Code of Transaction	560252
rrn	O	N(15)	Variable	RRN No for reference	104013006601

14 Appendix A (Transaction Status Code)

STATUS CODES	DESCRIPTION
50010	Init
50011	Capture Aborted
50012	3DS Start
50013	3DS Completed
50014	3DS Failed
50015	3DS Aborted
50016	Switch Start
50017	Switch Timeout
50018	Switch Aborted
50020	**Success**
50021	Failed
50097	Test Transaction

14 Appendix B (PG Error Codes)

PG ERROR CODES	Description
0	**No Error**
1	Call Issuer
2	Contact Switch Admin
3	Retry After Some Time.
10001	Disabled Instance
10002	Test Instance
10003	Instance under Maintenance
10004	Internal Server Error
10011	Disabled Acquirer
10012	Test Acquirer
10013	Acquirer under Maintenance
10021	Disabled Merchant
10022	Test Merchant
10023	Merchant under Maintenance
10024	Bad Input Data in Request
10032	Card Association not enabled
10033	Card Range not enabled
10040	Transaction not allowed - flow error
12001	Acquirer Server Error
12002	Acquirer Timeout
12003	Acquirer Down
12004	Acquirer Declined
12005	Batch Closed
13001	Issuer Server Error

13002	Issuer Timeout
13003	Issuer Down
13004	Issuer Declined
13005	Invalid Amount
13006	Issuer Insufficient Funds
15001	missing parameter
15002	invalid command
15013	amount error
15041	DECLINED(lost card)
15042	DECLINED (no account)
15043	DECLINED (stolen)
15051	no sufficient funds
15054	expired card
15055	wrong pin entered
15057	DECLINED (not cardholder)
15058	DECLINED (not terminal)
15059	fraud suspected
15061	DECLINED (exceeds with)
15062	restricted card used
15065	exceeded frequency
15091	Issuer or switch is inoperative
15092	no routing available
15096	system error/ PREVIOUSLY AUTHORIZED / PREVIOUSLY DECLINED
15110	no acct
15120	account closed
15399	system unavailable
15400	General error
15401	command is null or empty
15402	xml is null or empty
15406	not authenticated
15407	NOT AUTHORIZED
15408	xml data error
15409	shopper service error/Invalid AuthenticaionValue
15410	invalid bin or error with check bin
15412	IAS error
15413	Invalid OTP
15414	Expired OTP
15415	Exhausted OTP verification
15416	Exhausted OTP resend count
15417	Duplicate requestID

15 Appendix C (NPCI Error Codes)

A. Error Codes for Initiate2

Error Code	Description	Reference XML Tag
0/00	Transaction was successfully initiated and card details verified by issuer.	Entire SOAP envelop along with values of parameters passed BEPG will successfully complete all technical as well as business validations on XML structure and values of all the parameters as present within tag to return code '0'
01	Missing Parameter	**Entire SOAP envelop along with values of parameters passed** BEPG will check all the parameters and if any of the required parameter is not present then return this error code.
02	Invalid Command	**<strCommand>** BEPG checks if command in this tag is as per specs and if not will decline with this error code.
13	Amount Error	**<auth_amount> under <strXML>** If amount value send by PG is not greater than 0 then BEPG will decline with this error code.
96	System Error	In case if due to any technical error BEPG is unable to process transaction then BEPG to return this error code.
400	General Error	If BEPG is not able to process transactions for codes as specified in this document then BEPG can return this code.
401	Command is Null or Empty	**<strCommand>** if NULL or Empty command found, then decline with this error code
402	XML is Null or Empty	**<strXML>,** If XML body in this tag is Null or Empty then return this message
406	Not Authenticated	**<RequestorCredentials>, <partner_id>, <merchant_password>** Credentials mismatch
407	Not Authorized	**<RequestorCredentials>** **And Source IP** BEPG checks for credentials passed by PG and source IP and if source IP is not from white listed IP addresses at application level then BEPG will return this error code.

408	XML Data Error	BEPG will validate entire XML structure/format and if validation fails then BEPG will return this error code
410	Invalid BIN	BEPG will return this error code if BIN is not available or eligible for RuPay e-commerce transactions.
412	Issuer Authentication Failure	BEPG passes all card details to Issuer for verification and if verification is failed by Issuer, BEPG to send this error code.

B. Error Codes for Authorize , generateOTP API

Error Code	Description	Reference XML Tag
00	Transaction was successfully authorized by Issuing bank.	Entire SOAP envelop along with values of parameters passed BEPG will successfully complete all technical as well as business validations on XML structure and values of all the parameters present within tag and passes the same to Issuing bank. On successful response from Issuinb bank, BEPG to return code '00'
01	Missing Parameter	**Entire SOAP envelop along with values of parameters passed** BEPG will check all the parameters and if any of the required parameter is not present then return this error code.
02	Invalid Command	**<strCommand>** BEPG checks if command in this tag is as per specs and if not will decline with this error code.
07	Invalid Parameter	BEPG checks all parameters and if any parameter is not as per specs and it will decline with this error code.
13	Amount Error	**<auth_amount> under <strXML>** If amount value send by PG is not greater than 0 then BEPG will decline with this error code.
41	DECLINED (lost card)	**<strXML>** If card is marked as Lost at Issuer end, Issuer to pass this error code to BEPG and BEPG sends this error code to PG
42	DECLINED (no account)	**<strXML>** If no account is found for the card number provided at Issuer end, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.

43	DECLINED (stolen)	**<strXML>** If card is marked as Stolen at Issuer end, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.
51	NON SUFFICIENT FUNDS	**<strXML>** If sufficient funds are not available in cardholder account to process that transaction amount, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.
54	EXPIRED CARD	**<strXML>** If card is expired, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.
55	WRONG PIN	**<strXML>** If cardholder has submitted wrong PIN, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.
57	DECLINED (cardholder not allowed)	**<strXML>** If transactions are not allowed for the provided card at Issuer end, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.
58	DECLINED (terminal not allowed)	**<strXML>** If transactions are not allowed for the provided terminal details at Issuer end, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.
59	DECLINED (fraud)	**<strXML>** If transaction is match any RISK criterias/rules at Issuer end, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.
61	DECLINED (exceeds with)	**<strXML>** If daily withdrawal amount limit is exceeded for provided card at Issuer end, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.
62	DECLINED (restricted card)	**<strXML>** If card is marked as Restricted for usage at Issuer end, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.
65	DECLINED (exceeds frequency)	**<strXML>** If daily withdrawal frequency limit is exceeded for provided card at Issuer end, Issuer to pass this error code to BEPG and BEPG sends this error code to PG.

91	ERROR	If NPCI switch does not receives response from Issuer switch within specified time limit, transactions is declined by NPCI switch with this error code

92	NO ROUTING AVAILABLE	If no routing is available for provided BIN at NPCI switch end, NPCI switch to pass this error code to BEPG and BEPG sends this error code to PG.
96	SYSTEM ERROR	In case if due to any technical error BEPG is unable to process transaction then BEPG to return this error code.
96	PREVIOUSLY AUTHORIZED	Duplicate Authorize API Call ** Transaction was authorized at BEPG and received duplicate Authorize request for same transaction ID from PG. It is completely on PG's discretion to decline/approve the transaction based on response received for duplicate Authorize API call and member entity is always advised to trigger the Transaction Status API call in case of response not received for first authorize API call instead of duplicate Authorize API call.
96	PREVIOUSLY DECLINED	Duplicate Authorize API Call ** Transaction was authorized at BEPG and received duplicate Authorize request for same transaction ID from PG. It is completely on PG's discretion to decline/approve the transaction based on response received for duplicate Authorize API call and member entity is always advised to trigger the Transaction Status API call in case of response not received for first authorize API call instead of duplicate Authorize API call.
110	NO ACCT	If not account is found for provided card at Issuer end, Issuer to pass this error code to BEPG and BEPG to send this error code to PG.
120	ACCT CLOSED	If not account is closed for the card provided at Issuer end, Issuer to pass this error code to BEPG and BEPG to send this error code to PG.
399	SYSTEM UNAVAILABLE	In case if interfacing systems are unavailable at BEPG end, for some reasons resulting in non-processing of transaction then BEPG to return this error code
400	General Error	If BEPG is not able to process transactions for codes as specified in this document then BEPG can return this code.

401	Command is Null or Empty	**<strCommand>** IAS checks tag and if NULL or Empty command found, then decline with this error code
402	XML is Null or Empty	**<strXML>**, If XML body in this tag is Null or Empty then return this message

406	Not Authenticated	**<RequestorCredentials>, <partner_id>, <merchant_password>** Credentials mismatch/Invalid credentials OR not able to validate the signature/hash of the request in case or rest endpoints.
407	Not Authorized	**Source IP** BEPG checks for credentials passed by PG and source IP and if source IP is not from white listed IP addresses at application level then BEPG will return this error code.
408	XML Data Error	**<errorcode>** under **<strXML>** BEPG will validate entire XML structure/format and if validation fails then BEPG will return this error code
409	Invalid AuthenticaionValue	**The authenticationValue sent in request is validated and if invalid or duplicate, the request is rejected with this code.**
453	Exhausted OTP resend count	Limit reached for resend OTP request for this transaction. The user must initiate a new transaction.
454	Duplicate requestID	The requestID of generateOTP is already used for a different card number.
501	EMI Not available	EMI program is not available for the given card bin.
504	EMI details mismatch	The Emi details (emi tenure, emi rate of interest) does not match with EMI program available for the card bin.
505	SI not registered	SI registration ID sent in request is not available in the system.
506	Invalid relationshipID	The relationshipID sent in request is not available in the system or is invalid
507	SFA device token security error	The deviceToken is invalid for the relationshipID sent. Note: Reason could be 1. deviceToken decryption error 2. deviceToken signature error 3. devicetoken validation error
508	SFA token security error	The token/token signature is invalid

C. Error Codes for Verify OTP

Error Code	Description	Reference
00	Request was successful.	
01	Missing Parameter	BEPG will check all the parameters and if any of the required parameter is not present then return this error code.
07	Invalid Parameter	BEPG checks all parameters and if any parameter is not as per specs and it will decline with this error code.
13	Amount Error	If amount value send by PG is not greater than 0 then BEPG will decline with this error code.
406	Not Authenticated	Credentials mismatch. Invalid credentials or not able to validate the signature/hash of the request.
407	Not Authorized	BEPG checks for credentials and source IP and if source IP is not from whitelisted IP addresses, then BEPG will return this error code.
450	Invalid OTP	Invalid OTP. OTP Authentication failed.
451	Expired OTP	The OTP has expired. User must ask for a fresh OTP.
452	Exhausted OTP verification	OTP verification count exhausted for this transaction. User must initiate a new transaction.
501	EMI Not available	EMI program is not available for the given card bin.
502	Amount not eligible for EMI	EMI program available for the given card bin but the amount is not eligible for EMI conversion.
503	Card not eligible for EMI	EMI program available for the given card bin but the card is not eligible (issuer rejection)

16 PAYLOAD

REQUEST POSTING

1) Request should be posted in ***application/json,***response will be sent in ***application/json***.
2) Headers must be passed in request for request verification purpose as shown below, all the header values will be provided from PG.

3) Request must be posted to the respective API endpoint which will be provided byPG.

```
"Request Payload":{
     "param1” : “value1”,
     “param2” : “value2”
}
```

CHECKBIN

```
"Request Payload":{
    “pgInstanceId”:”16787434”,
    "cardBin":“652851000”
  }
}

Response Payload:{
     "status":"success",
     "errorcode":0,
     "errormsg":"",
     "qualifiedInternetpin":"TRUE",
     "authenticationNotRequired":"FALSE",
     "availableAuthMode":"02",
     "additionalProductsSupported":"01,03"
  }
```

INITIATE

```
"Request Payload":{
    "pgInstanceId":”720200”,
    "merchantId":”123”,
    "pan": “XXX607512312344412XXX”,
    "expDate”:"022024",
    "cvd2":”123”,
    "nameOnCard":”test”,
    "email":”test@test.io”,
    "currencyCode":”356”,
    "amount":”11025”
    "merchantReferenceNo":”122344sdg”
    "orderDesc":”One bat”,
    "mobileNumber":”8867772222”
    "httpAccept": “text/html, application/xhmtl (http accept of a browser)”,
     "browserUserAgent":"user agent of a browser",
     "ipAddress":"ip-address of a customer",
     “ext1”:”123456”,
     “ext2”:”123456”,
     “amountInINR:”11025”,
     “originalAmount”:”11025”,
      “merchantResponseURL”:”https://merchantdomain.com/response”
     “purposeOfAuthentication”:” Real Card Transaction”,
     “tokenAuthenticationValue”:”wwopqds/qww=”
  }
}




Response Payload:

{
   "status": "success",
   "redirectURL":
"https://qccert.npci.org.in/bepg/cert/iasAuthentication?AccuCardholderId=27632769139&A
ccuGuid=8d82faff-35ee-43d4-9dba-4d49ffc8ca6d&AccuHkey=a30d00ee-74d5-47dd-ae61-
db6c58f208f6",
   "accuRequestId":
"MzM3YjdjZjdlNjA2ZTlkY2Y2NTUzOTI4NmNjM2NiNzdlMzg4ODM5YmY4NTFmODBjZmJhODFlN
DU4NTc3MDE0Yw==",
   "session": "sendRupayRedirectionResponse|641646|PayU02|10003862|16479914",
   "accuReturnUrl": "http://localhost:9003/web/v1/parseRupayResponse",
   "pgTransactionId": "641646",
   "errorcode": 0,
   "errormsg": "",
   "rupayTransactionId": "10011202210250000000000186019"
}
```

Authentication Response(Form Post)

```
Response Payload:{
     "status":"success",
     "errorcode":"0",
     "errormsg":"",
     "pgTransactionId":"32424"
  }
```

Generate OTP

```
"Request Payload":{
   "pgInstanceId":”720200”,
   "merchantId":”123”,
    "pan": “XXX607512312344412XXX”,
    "expDate_:"022024",
    "cvd2":”123”,
    "nameOnCard":”test”,
    "email":”test@test.io”,
    "currencCode":”356”,
    "amount":”11025”
    "merchantReferenceNo":”122344sdg”
    "orderDesc":”One bat”,
    "mobileNumber":”8867772222”,
    “cardHolderStatus”:”NW”,
    "http_accept": “text/html, application/xhmtl (http accept of a browser)”,
    "browser_user_agent":"user agent of a browser",
    "customerIpAddress":"ip-address of a customer",
    “ext1”:”123456”,
    “ext2”:”123456”,
    “amountInINR:”11025”,
    “originalAmount”:”11025”
    “purposeOfAuthentication”:”Token Transaction”,
    “tokenAuthenticationValue”:”wwopqds/qww=”
  }
}

Response Payload:{
     "status":"success",
     "errorcode":"0",
     "errormsg":"",
     "pgTransactionId":"32424",
     “validityPeriod”:”25”
   }
```

Resend OTP

```
"Request Payload":{
   "pgInstanceId":”720200”,
   "merchantId":”123”,
   “merchantRefernceNo”:”dafa34”,
   “pgTransactionId”:”1232324df”
  }
}



Response Payload:{
     "status":"success",
     "errorcode":"0",
     "errormsg":"",
     "pgTransactionId":"32424",
     “validityPeriod”:”25”
   }
```

Verify OTP

```
"Request Payload":{
   "pgInstanceId":”720200”,
   "merchantId":”123”,
   “merchantRefernceNo”:”dafa34”,
   “otp”:”108434”
    “pgTransactionId”:”1232324df”
 }
}



Response Payload:{
     "status":"success",
     "errorcode":"0",
     "errormsg":"",
     "pgTransactionId":"32424"
  }
```

Authorize

```
"Request Payload":{
   "pgInstanceId":”720200”,
   "merchantId":”123”,
   “merchantReferenceNo”:”dafa34”,
   “pgTransactionId”:”1232324df”
 }
}



Response Payload:{
     "status":"success",
     "errorcode":"0",
     "errormsg":"",
     "pgTransactionId":"32424",
     “rupayTransactionId”:”10000000002332”
  }
```