Aerion PG Transaction Service

API Document

For Merchants / Payment Aggregators

Version 1.1

Last Updated:

22nd August 2023

Revision History

Version Number	Modification Made	Modified By	Reviewed By	Date
1.0	Initial Draft	Vinod	Bhaumik	28th April 2022
1.1	Message Level Encryption	Bhaumik	Ravi	12th Jan 2023
1.2	Added ALT ID changes	Amritha	Ramesh	22nd August 2023

Table of Contents

INTRODUCTION

This document explains about the API's that merchants / payment aggregators / sub clients need to develop and implement at their side to integrate with Wibmo's Aerion PG.

PURPOSE OF DOCUMENT

This document explains API spec that is listed below

- API gateway API
- SALE
- PRE AUTH
- PRE AUTH-COMPLETE
- PRE AUTH-CANCEL
- SI Registration and Recurring
- Void/Refund transactions.
- Transaction Status API

REQUEST FORMAT FOR API GATEWAY

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	Merchant id

RequestBody

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	Merchant id
clientReferenceId	O	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+wOJ3Eo6l4RVgIEru/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
errorMessage	M	Error Message
var1	O	
var2	O	
var3	O	

Sample Response


```
{
```

```
    "signedEncResponsePayload":
"LTPpSHNwAuEthTTDlmO0G9QY5dKfT7COTFaJe+7FhKkrgVYrSkJIxQeFWhi0tEghUDm54yE4QokEk42ifRpenCcGC1fLF+ReOsErEhCEHy1mUUE+34soWWdhX
qEY3GflWyJK6v7sr1A6UVKgJCraSpJy5Ubhzrxtt10Jh/bojo6NJysRQszgEnYUagzDqW5YVIjgG9InJ/2meJpQt3aCx9scgHG6A8iJBqNmaJYg1cjB4KVRZGF
zRSu3TKF3miSmwG7CU1qC8+9F+DyD2ExQV9/8puN5bpVlJkiOnye1GGzyBh3LXXGYDS7yaKoR6joGsPuFrJN48uZy9UqrS/qvaCo9/zGhh2gZEe1qxGNxV649A
W5pdQMdrvRGgYRghioAOtgLJneSnADjO5L4xWEn1wnY/QjTxpCPh52n4K7PxtjxPS6wOTZaqOUY32+fn7Fn3O3pjBr5go6Dasynkl2UeRs4obc5ReUD3AwSonv
qVAtLCyApmmCUAHZloCWVd20NHubzQRPJHPmHjx+bmTn6etPO2w/rS4Y1KhGlSQ5WzDUJzYh3MMOhV7aKXTeKmbxLuEKQebZl8ajEIv3DKLZjAk5cmuFk8el8T
uiTpsXTxnqR7Azapc2JZHZBBoMvhcpW/1ACnbsdI3vc9kltEQThRWXqhuyoIMdijUifUmNAScStbZMMPGbNdx1tey313YiP6awakCCHyEyFHsmofOPQRMxxoMe
5z32UF1Ky/mzVOde1Abn8ZLJuFw/NI7fSYLfYTKQxklGe+vjmB0czti96TuUc0A9rSFhvXJUjjwQpiuzLbT9fmQjzh+WLl7lVSf7Tc21bePN4ZYQv9jF24SVeN
8jsMrtUXgkYShucUmNDcpiBzCyWYjKH/HG1MMGhhEKpBo7xScrdqZ8k7/8jf1mXBqwe+CwHBjewCV55GUaTBJBhorv5ftIqQCNvvDXDvWeEa/OJ2f/Ze8oNzi1
KcRlKkFhIqzQAuUUCAQIJ3wnIwfBlpRXXqbj4mK2M0/CUoMUKpARe+VLJFmF7kPmp36gaY2oi2cFwmtshPGX/JZGR9va5u2Rps329AoA1DMyvsh0SoOcZp54X8
qYKg8EXlG+TLVlqcOc4fN4oSwMkV8JvT1s0nu7/bSNR6afRI12omVxYml3VljKoGNRCgMN2R+LYi5ziOvFm9kO4IdF1vjQhAqxH8TvuqCX2zgqjNwd/1Nbwfx2
wl/ufMd/BsjkbUOQKt0CZTt8y7vYGh3gCMCpDxIG1dirLED+liVFaZybzhZhQUHG3otiDNhDfjQJn/3B0CKG6YCdHJjUg",
    "symmetricEncKey":
"iPeXe2KXche23UZHURwFw+BdORL5xSPp66wXVJfAQRNcKR+nnwZ4dOsAKJX3dmSQ0lx6Cp6MpZBZo2UQH2eQJbt1SJmFK2pPFNXD+L7wYcBs/p4xg4xk8l7HK
7E1qkxoJEsKS9DunqYvl4IjW/GVS9G2+rHSEB2VrRlP4Q3k++Tb8Rp94nFjSv6wYIz4ghZdLllqnbw056rAG9mOF3i00box5kGZdX4xG+oLjT/GSzi56mx2Svc
tNXmiB4N9+6BC7GxbkCc4lLYhRa9MDTPcoz1HkF1Cjcp+ldYGZsYf3KllfmmrVeIlBq8F9ABUzwZjot3GE12gX4EdZSnRryabfA==",
    "iv": "Ob06nKjivp+plDCY",
    "clientReferenceId" : "aaddvkkllle123344”,
    "statusCode": "PG99200",
    "errorMessage": null,
    "var1": null,
    "var2": null,
    "var3": null
}
```


Error Codes

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

REQUEST FORMAT FOR SALE AUTHORIZATION

POST: https://pgdomain/api/v1/sale
content-type: application / json

Decrypted Body

*M/O = Mandatory/Optional/Conditional

Field Name	M/O / C	Description	Example
pgInstanceId	M	PG instance ID	72702415
merchantId	M	PG Merchant ID	35890949
acquiringBankId	M	PG Acquiring Bank ID	93734895
action	M	Action To Be Performed	SERVICE_POST_MPI SERVICE_POST_MPI_SI SERVICE_POST_MPI_RECURRING
transactionTypeCode	O	Code of transaction types Mandatory for Action SERVICE_POST_MPI Optional for other actions	9003 for Sale 9001 for Pre Auth
deviceCategory	O	0 for Web,	0
pan	M	Should be a valid Visa/Master card, all the digits as a continuous string without formatting ALT ID to be passed for domestic guest checkout transactions	Original Values Visa: 4123123412341234 MC: 5113123412341234
expiryDateYYYY	M	Should be a valid card expiry year in the format yyyy ALT ID expiry to be passed for domestic guest checkout transactions	Original Value 2010
expiryDateMM	M	Should be a valid card expiry month in the format mm ALT ID expiry to be passed for domestic guest checkout transactions	Original Value 12
cvv2	O	Should be a valid cvv2 or cvc2	Original Value 123
nameOnCard	M	Should be a valid name on card	Govind S
email	O	Should be a valid email	mail@gmail.com
currencyCode	O	ISO Code for the currency of amount field. For INR this is 356. If not passed default currency would be used. Len: 3 char.	356
amount	M	Purchase amount including the minor units of currency with all punctuation removed i.e., implied decimals and no formatting	100
merchantReferenceNo	M	Merchant reference number	AX143565
orderDesc	M	Brief description of items purchased. Len: 0-50 chars	Apple iPOD 20GB
customerDeviceId	O	Unique ID that identifies customer device used – IP or Mobile Number or MAC or IMEA Len: 0-30 char (my be empty)	+919123412345
mpiTransactionId	C	Transaction ID from MPI Mandatory for pre auth and sale. Optional for recurring transactions.	1234
threeDsStatus	C	Status from MPI Mandatory for pre auth and sale. Optional for recurring transactions.	Y

threeDsEci	C	ECI from MPI Mandatory for pre auth and sale. Optional for recurring transactions.	05
threeDsXid	C	XID from MPI (length 28 – encoded value) For v1.O Mandatory for pre auth and sale. Optional for v2.O and for recurring transactions. non Empty string need to be passed.	QUJDREVGR0hJSjEyMzQ1Njc4OTA=
threeDsCavvAav	C	CAVV or AVV from MPI (length 28 – encoded value)	AAABBGkgUhI0VngQACBSAAAAAA=
messageHash	O	This is a hash of the fields sent to ensure that no modification is done in transit between Merchant/PG and web servers. ***Composition of message_hash for service post mpi, preauth and si registration*** SERVICE-POSTMPI:17:Base64(SHA1(pg_instance_id|merchant_id|action| currency_code|amount|merchant_reference_no|pan|exp_date_yyyyexp_date_mm|cvv2|name_on_card|customer_device_id|mpiId|status|eci|xid|cavv|key|)) ***Composition of message_hash for service post recurring*** SERVICE-POSTMPI:17:Base64(SHA1(pg_instance_id|merchant_id|action| currency_code|amount|merchant_reference_no|name_on_card|siUniqueRefNum|key|)) When the PG receives the request, it follows the same process as above to re-generate the hash and check it with the message_hash field. An exact match ensures that the message has not been modified in transit. Note: PG will provide a software kit for hash generation and verification.	See description. Note: PG will provide a software kit for request generation (including hash) and verification. No need to code for it.
ext1	O	This field captures ctx id, which is unique for every transaction.	123456
ext2	O	This field captures mtx id	123456
ext3	O	Merchant can pass any additional data(specific to merchant) not included in the message hash	- do -
ext4	O	Merchant can pass any additional data(specific to merchant) not included in the message hash	- do -
ext5	O	Merchant can pass any additional data(specific to merchant) not included in the message hash	- do -
ext6	O	Merchant can pass any additional data (specific to merchant) not included in the message hash	- do -
ext7	O	Merchant can pass any additional data (specific to merchant) not included in the message hash	- do -
ext8	O	Merchant can pass any additional data(specific to merchant) not included in the message hash	- do -
ext9	O	Merchant can pass any additional data(specific to merchant) not included in the message hash	- do -
ext10	O	Merchant can pass any additional data(specific to merchant) not included in the message hash	- do -

amounInInr	O	Used to capture the base currency amount i.e INR for international transactions in implied decimals.	20000
trid	O	Token Requestor ID, Only for Tokenized transactions if it is available.	123456789
cryptogram	O	Cryptogram value to be passed, which is received from Token Vault for only Tokenized transactions. Cryptogram is mandatory for Token transactions and ALT ID transactions. Cryptogram is optional for Subsequent recurring transactions.	Visa - AgAAAGQDrmEx5r0AmbHTg0AAAAA= MC - "11223344556677889900112233445566778899"
threeDsVersion	O	Authentication completed in 1.0 or 2.0	1 for 1.0, 2 for 2.0
threeDsTxnId	O	Directory Server Txn ID received from 3DSS Server. Mandatory for 2.0 txn	12345789
installmentFrequency	O	indicates the frequency for paying the installments.	'DAILY','WEEKLY','FORTNIGHTLY','MONTHLY','QUARTERLY','HALFYEARLY','YEARLY','ADHOC'
maxDebitAmount	O	Indicates the maximum amount which can be debited for each installment/recurring payments i.e., amount to be stored with implied decimals	30000
siUniqueRefNum	O	Unique reference number used to uniquely identify SI registration and recurring payments	SI143565
installment	O	Indicates the maximum number of permitted authorizations for installment/recurring payments. Len: 0-3 char and must be > 1(if not empty)	4
postalCode	O	Postal code/Zip to be passed. Len – MAX 9 AN	560060
streetAddress	O	Street address data to be passed. Len – max 50 AN	98 West Airport Avenue Newton, NJ 07860
shippingAddress	O	Cardholder shipping address data to be passed. Len – max 40 AN	8710 Edgemont Street Englishtown, NJ 07726
altIdFlag	C	This field is Mandatory for Domestic guest checkout transactions	Y/N

RESPONSE FORMAT FOR SALE AUTHORIZATION

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

Field Name	M/O/C	Description	Example
transactionId	M	Unique value generated for each transaction internally by the PG	123456
status	M	Transaction Status	50020 = SUCCESS (any other value is a failure, full status codes shall be provided in appendix A)
pgErrorCode	M	PG Error Code	0 = No Error (any other value means some kind of error, full error codes list shall be provided in appendix B)
pgErrorDetail	M	PG Error detail	pg error detail was not set
orderDesc	M	Brief description of items purchased. Len: 0-50 chars	Apple iPOD 20GB
merchantReferenceNo	M	Merchant reference number	AX143565
approvalCode	C	Approval code of Transaction	560252
rrn	O	RRN no for reference	104013006601
creditDebitCardFlag	O	Flag to say whether its credit card or debit card	'C' – Credit card 'D' – Debit Card 'P' – Prepaid Card
ext1	O	Same value will be sent back as pg has received from merchant in the request.	
ext2	O	Same value will be sent back as pg has received from merchant in the request.	
ext3	O	Same value will be sent back as pg has received from merchant in the request.	
ext4	O	Same value will be sent back as pg has received from merchant in the request.	
ext5	O	Same value will be sent back as pg has received from merchant in the request.	
ext6	O	Same value will be sent back as pg has received from merchant in the request.	
ext7	O	Same value will be sent back as pg has received from merchant in the request	
ext8	O	Same value will be sent back as pg has received from merchant in the request.	
ext9	O	Same value will be sent back as pg has received from merchant in the request.	
ext10	O	Same value will be sent back as pg has received from merchant in the request.	

PRE AUTH-CAPTURE/CANCEL TRANSACTION FLOW

```
/api/v1/preauth
```

For orders which get successfully pre-authorized, consumer's card limit is booked to the value of order amount. But the card is actually debited only after the merchant send Pre Auth Completion request to PG and subsequently settles it. To achieve this Merchant has to call the Pre Auth Completion/Capture API as per the below mentioned spec.

If Merchant wants to cancel the Pre Auth transaction, then Merchant has to call the Pre Auth Cancel API, then PG will mark the pre Auth transaction as failed. Post that If we try to Capture the Pre Auth transaction, it will be declined at PG end.

REQUEST FORMAT FOR PRE-AUTH COMPLETION(CAPTURE) & CANCEL

HEADERS

Name	Value (Example)	Description
x-api-key	777a9184-2e70-11ed-b928-005056b59d84	Actual api-key will be provided from PG during onboarding

BODY

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

Html Field Name	M/O/C	Description	Example
pgInstanceId	M	PG instance ID	79070160
merchantId	M	PG Merchant ID	44065131
action	M	Action To Be Performed	SERVICE_POST_MPI_PRE_AUTH_COMPLETION SERVICE_POST_MPI_PRE_AUTH_CANCELLATION
amount	M	Purchase amount including the minor units of currency with all punctuation removed i.e., implied decimals and no formatting	10000
originalTxnId	M	Transaction ID where pre auth have been approved based on this transaction id the capture transaction occurs	32891

RESPONSE FORMAT FOR PRE-AUTH COMPLETION(CAPTURE) & CANCEL

Field Name	M/O/C	Description	Example
transactionId	M	Unique value generated for each transaction internally by the PG	123456
status	M	Transaction Status	50020 = SUCCESS (any other value is a failure, full status codes shall be provided in appendix A)
pgErrorCode	M	PG Error Code	0 = No Error (any other value means some kind of error, full error codes list shall be provided in appendix B)
pgErrorDetail	M	PG Error detail	pg error detail was not set
orderDesc	O	Brief description of items purchased of pre auth. Len: 0-50 chars	Apple iPOD 20GB

merchantReferenceNo	O	Merchant reference number of pre auth	AX143565
approvalCode	O	Approval code of Transaction of pre auth	560252
rrn	O	RRN no for reference of pre auth	104013006601
creditDebitCardFlag	O	Flag to say whether its credit card or debit card of pre auth	'C' – Credit card 'D' – Debit Card 'P' – Prepaid Card
ext1	O	Same value will be sent back as pg has received from merchant in the request of pre auth.	
ext2	O	Same value will be sent back as pg has received from merchant in the request of pre auth.	
ext3	O	Same value will be sent back as pg has received from merchant in the request of pre auth.	
ext4	O	Same value will be sent back as pg has received from merchant in the request of pre auth.	
ext5	O	Same value will be sent back as pg has received from merchant in the request of pre auth.	
ext6	O	Same value will be sent back as pg has received from merchant in the request of pre auth.	
ext7	O	Same value will be sent back as pg has received from merchant in the request of pre auth	

REQUEST FORMAT FOR void/refund/void or refund

POST: https://pgdomain/api/v1/voidorrefund
content-type: application / json

HEADERS

Name	Value (Example)	Description
x-api-key	777a9184-2e70-11ed-b928-005056b59d84	api-key will be provided from PG during onboarding

BODY

*M/O = Mandatory/Optional/Conditional

Field Name	M/O/C	Description	Example
pgInstanceId	M	PG instance ID.	79070160
merchantId	M	PG Merchant ID.	44065131
action	M	Action to be performed.	voidorrefund
originalTransactionId	M	Original transaction id of the auth.	1234
merchantReferenceNo	M	Merchant reference number of the authorized transaction on which the void/refund to be done.	AX14356
messageHash	O	This is a hash of the fields sent to ensure that no modification is done in transit between Merchant/PG and web servers. ***Composition of messageHash for voidorrefund service*** voidorrefund:7:Base64(SHA1(pgInstanceId|merchantId|action|originalTransactionId|amount|merchantReferenceNo|key|)) When the PG receives the request, it follows the same process as above to re-generate the hash and check it with the messageHash field. An exact match ensures that the message has not been modified in transit. Note: PG will provide a software kit for hash generation and verification.	See description. Note: PG will provide a software kit for request generation (including hash) and verification. No need to code for it.
amount	M	Pass amount to refund either full or partial refunds to be done by the merchant. Based on the merchant refund parameters, those many times refunds will be allowed.	10000 (need to send the amount in implied decimals. E.g.1.00 = 100) Max – equals to sale amount min – 1 Note: For Void request full auth amount needs to be passed.
refundType	O	This parameter is mandatory for Surcharge Merchants. Identifying whether the full refund needs to be given to customer or not based on the flag which includes reverting the surcharge's / Service Tax etc... **CN - Only amount is refunded to customer which is passed in the request, no surcharge is reversed to customer RF - Amount + Surcharge + any Taxes leveraged during the settlement then entire amount needs to be returned to the customer Note : Correct amount needs to be sent by the merchant with its refund type.**	CN/RF Values may differ based on the integration Note: RefundType is optional for void transactions.
ext1	O	This field accepts Citrus Transaction Id from payU	CTX147258369QAZXWERTAXSFGHQWED Note: ext1 is optional for void transactions.

RESPONSE FORMAT FOR REFUND

content-type: application / json

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

Field Name	M/O/C	Description	Example
status	M	Transaction Status	50020 = SUCCESS (Any other value is a failure, full status codes shall be provided in appendix A)
pgErrorCode	M	PG Error Code	0 = No Error (Any other value means some kind of error, full error codes list shall be provided in appendix B)
pgErrorDetail	M	PG Error detail	pg error detail was not set
transactionId	M	New Transaction id generated by PG	1235
rrn	M	Retrieval reference number	06711005310

REQUEST FORMAT FOR TRANSACTION STATUS ENQUIRY

POST: https://pgdomain/api/v1/txn-status
content-type: application / json

*M/O = Mandatory/Optional/Conditional

Field Name	M/O/C	Description	Example
pgInstanceId	M	PG instance ID	79070160
merchantId	M	PG Merchant ID	44065131
currencyCode	O	ISO Code for the currency of amount field. For INR this is 356. If not passed default currency would be used. Len: 3 char.	356
amount	M	Purchase amount including the minor units of currency with all punctuation removed i.e., implied decimals and no formatting	100
merchantReferenceNo	M	Merchant reference number	AX143565
messageHash	O	This is a hash of the fields sent to ensure that no modification is done in transit between Merchant/PG and web servers. *Composition of message_hash* CURRENCY:7:Base64(SHA1(pgInstanceId|merchantId|currencyCode|amount|merchantReferenceNo|key)) When the PG receives the request, it follows the same process as above to re-generate the hash and check it with the message_hash field. An exact match ensures that the message has not been modified in transit.	

RESPONSE FORMAT FOR TRANSACTION STATUS ENQUIRY

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

Field Name	M/O/C	Description	Example
transactionId	M	Unique value generated for each transaction internally by the PG	123456
status	M	Transaction Status	50020 = SUCCESS (any other value is a failure, full status codes shall be provided in appendix A)
pgErrorCode	M	PG Error Code	0 = No Error (any other value means some kind of error, full error codes list shall be provided in appendix B)

pgErrorDetail	M	PG Error detail	pg error detail was not set

REQUEST EXAMPLE

```
{
  "pgInstanceId": "24730649",
  "merchantId": "71389820",
  "amount": 1000,
  "merchantReferenceNo": "UniqueMRN",
  "messageHash": "CURRENCY:7:bcdfx-=",
  "currencyCode": "840"
}
```

RESPONSE EXAMPLE

```
[
  {
    "transactionId": 27,
    "status": 50020,
    "pgErrorCode": 0,
    "pgErrorDetail": "No Error"
  },
  {
    "transactionId": 34,
    "status": 50020,
    "pgErrorCode": 0,
    "pgErrorDetail": "No Error"
  }
]
```

APPENDIX A (TRANSACTION STATUS CODES)

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

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
10005	Invalid Data Sent to Switch
10006	Internal Error caused contact Switch Admin
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
10025	PGInterface not allowed
10026	Merchant velocity check failed
10027	Merchant transaction country restriction
10030	Capture Aborted
10031	Auth Aborted
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
12006	Totals Mismatched
12007	Unable to settle
13001	Issuer Server Error
13002	Issuer Timeout
13003	Issuer Down
13004	Issuer Declined
13005	Invalid Amount
13006	Issuer Insufficient Funds
14001	3DS Failed
14002	3DS Aborted
14003	MPI Error