Amazon Pay Server-Side Integration Guide

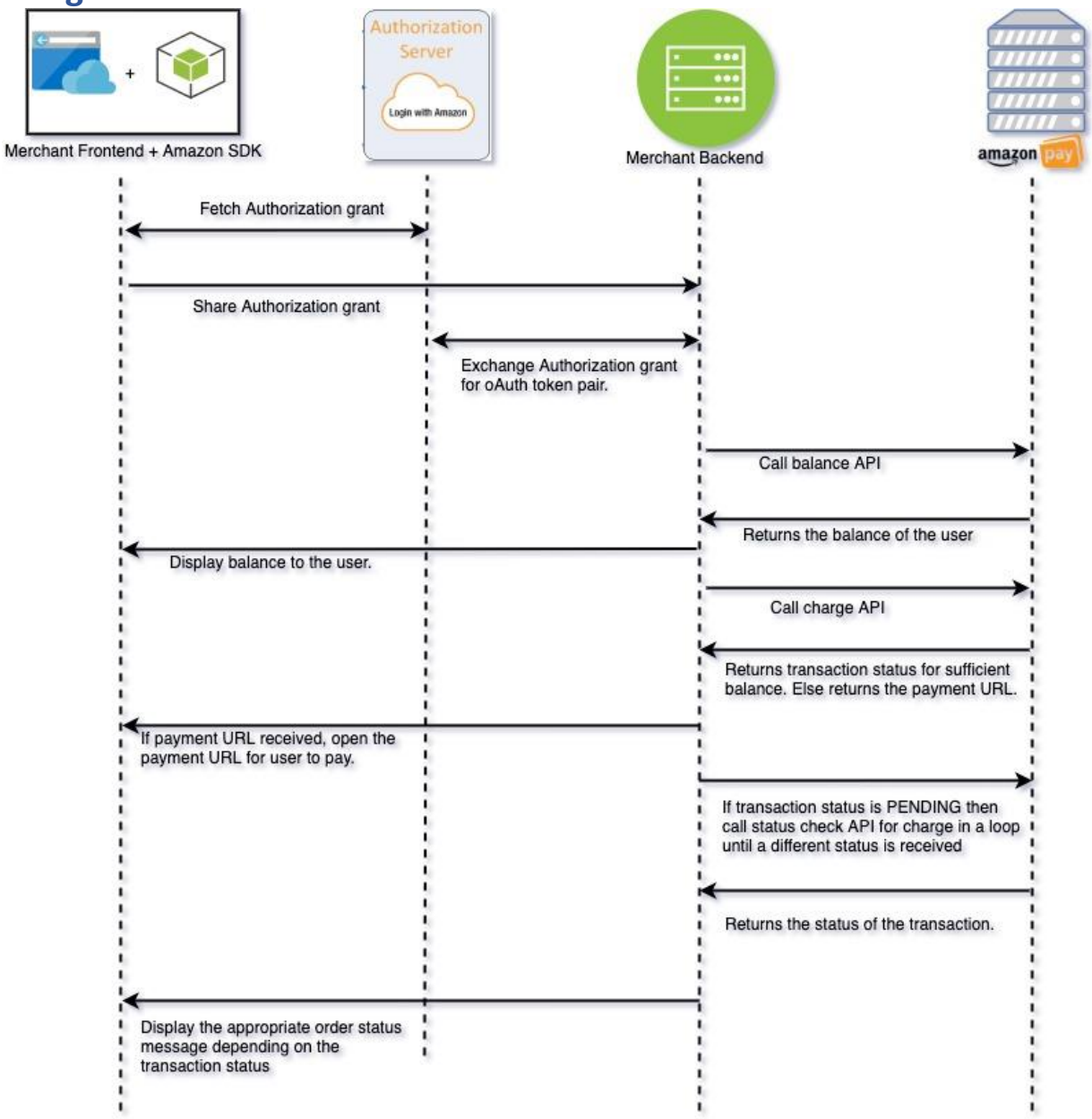
API documentation – V6

Amazon Pay API Integration Guide

Introduction

Amazon provides APIs to integrate Amazon Pay India on the 3rd party merchant site/Apps. These APIs can be used on the server-side to get balance details and process transaction for a user thus bringing the processes to the backend of the merchant and reducing the dependency on client side.

Integration Flow:



High level workflow

Integration Steps:

Account Linking:

1. Call Amazon Pay's client-side SDK when the user clicks on the Link account button on the App.
2. Client-side SDK will invoke the Login and consent page for the user.
3. Once a user gives consent then you will receive an Auth code that has to be used to get the access token of the user using the server-side API.

> **NOTE: Amazon Pay pages opening inside in-app webview on Apps is non-complaint with our infosec policy. Please read the client-side SDK documentation to understand the complete requirement.**

Fetching various Instruments:

1. Call the Instruments API to fetch Amazon Pay Balance (APB) of the user and also user's other stored payment methods.
2. You can then display all the options to the user to choose from.

Payment:

1. Call the Charge API with intent as ***Capture*** to initiate a transaction.
2. Check the status of any transaction by calling the Get Charge API with *txn_type = charge*.

Top-Up:

1. Take amount as an input from the user with which he/she wants to top-up their Amazon Pay balance.
2. Call the TopUp API with the requested amount.
3. Call the Get top-up API with *txn_type = topup* to get the status of the topup by the user.
4. Call the Balance check API to display the updated balance to the user.

Refunds:

1. You can refund an amount by calling the Refund API.
2. You can check the status of a refund by calling the Get Refund API with *txn_type = refund.*

Server-Side API Reference

Endpoints:

Sandbox endpoint: **https://amazonpay-sandbox.amazon.in**
Production endpoint: **https://amazonpay.amazon.in**

Request Header for all APIs (except token API):

Parameter	Type	Description
x-amz-client-id	String	Merchant ID provided by Amazon.
x-amz-source	String	The source of transaction origination like Android, iOS, Browser, Server.
x-amz-user-ip	String	User's ip from where transaction is originating
x-amz-user-agent	String	The user agent of the browser used by the user or the device info.
x-amz-algorithm	String	The algorithm used for signature generation. Possible values - ***AWS4-HMAC-SHA384***
Authorization	String	AMZ <access_key_id>:<signature> e.g. AMZ 44CF9590007:jZNOcbfWmD/A/f3hSvVzXZjM2HU=
x-amz-date	String	Date must be in UTC and should be same as the date used during signature generation. Ex 22010101T204041Z
x-amz-expires	String	Provides the time period in seconds, for which the generated signature is valid.

Response header for all APIs (except token API):

Parameter	Type	Description
x-amz-request-id	String	Unique Id provided by Amazon for each request.
x-amz-signature	String	Signature generated by Amazon for you to validate the response.

API Reference:

Access Token:

This API returns a token pair called access_token and refresh_token. Access token has to be used in all the other APIs as a user identifier.
There are two ways to call this API -

1. Using the Auth grant from the client-side SDK.
2. Using the refresh token if you already have the refresh token.

Method: POST
Path: https://api.amazon.co.uk/auth/o2/token
ContentType: application/json

Request Parameters:

Body: (Using Auth grant for request via Mobile Apps)

Parameter	Type	Description
grant_type	String	This is a constant ***authorization_code***
code	String	The Auth Code returned by the client-side SDK or the refresh token.
client_id	String	The Client ID returned by the client-side SDK

code_verifier	String	The Code Verifier passed to the client-side SDK while requesting Auth code. Code Verifier is the random code generated before SHA256 encryption and base64 encoding.
redirect_uri	String	The redirect URI returned by the client-side SDK

Body: (Using Auth grant for request via website)

Parameter	Type	Description
grant_type	String	This is a constant ***authorization_code***
code	String	The Auth Code returned by the client-side SDK or the refresh token.
client_id	String	The Client ID returned by the client-side SDK
client_secret	String	The secret value assigned to the client during registration.
redirect_uri	String	The redirect URI returned by the client-side SDK

Body: (Using refresh token)

Parameter	Type	Description
grant_type	String	This is a constant ***refresh_token***
refresh_token	String	The refresh token returned by the original Access token response.
client_id	String	The Client ID returned by the client-side SDK
client_secret	String	The secret value assigned to the client during registration. [**Only for Tokens obtained from web tokenization. No need to pass this for Apps**]

Response Parameters:

Parameter	Type	Description
access_token	String	The access token for the user account. Maximum size of 2048 bytes.
token_type	String	The type of token returned. Should be returned as ***bearer***.
expires_in	String	The number of seconds before the access token becomes invalid.
refresh_token	String	A refresh token that can be used to request a new access token. Maximum size of 2048 bytes.

Sample:


```
{
  "access_token":"Atza|IQEBLjAsAhRmHjNgHpi0U-Dme37rR6CuUpSR",
  "token_type":"bearer",
  "expires_in":3600,
  "refresh_token":"Atzr|IQEBLzAtAhRPpMJxdwVz2Nn6f2y-tpJX2DeX"
}
```


If you get this response instead then show the link account button to the user again.


```
{
“error_description”: “The request has an invalid grant parameter : refresh_token”,
“error”: “invalid_grant”
}
```


Instruments:

This API returns one or more payment instruments depending on the request. You can pass instrument type to get the required details. You must display the Amazon Pay Balance and other details to the user on your payments UI.

Method: GET
Path: <endpoint>/v1/payments/instruments

Request Parameters:

Query Parameters:

Parameter	Type	Description
merchantId	String	Your merchant id with Amazon Pay
accessToken	String	Access token fetched for the user.
instrumentTypes	String	Types of user’s payment instrument with us. Currently it supports only user’s balance enquiry.

Instrument Types:

value	Description
AmazonPayBalance	For fetching the balance of the user.

Request Sample:

```
https://{{endpoint}}/v1/payments/instruments?merchantId=A28RUURDCTQXU1 &accessToken=Atza|IgEBdsfsdfdsf&instrumentTypes=AmazonPayBalance
```

Response Parameters:

Sample:


```
{
  "amazonPayBalance": {
    "enabled": true,
    "priority": 0,
    "totalBalance": "7675.34",
    "redeemableBalance": "7675.34"
  }
}
```


NOTE: redeemableBalance is the balance you must display to the user.

Charge:

This API is used to initiate the transaction by passing the intent is ***Capture*** then the user will be instantly charged.
In case user has sufficient balance, the amount will be deducted directly. In case user has in-sufficient balance a payment url will be returned to which the user should be redirected to complete the payment.

Method: POST
Path: <endpoint>/v1/payments/charge
ContentType: application/json

Request Parameters:

Body Fields:

Parameter	Type	Description
intent	String	This is to identify the type of charge request. Possible values – ***Capture***.
amount	String	Contains the amount to be charged for the order. You can only send up to 2 digits after decimal point.
currencyCode	Sting	Will always be ***INR***
callbackUrl	String	URL where the user will be redirected back post payment completion in the case of multi-tender payment. We need this for **whitelisting** at our end and the URL should be a static one. For transaction via Apps the URL has to be specific to our requirements that can be found under Appendix section.
accessToken	String	This is the access_token generated for the user. It will be used to identify the user at our end.

chargeId	String	Unique id generated by merchant for every transaction. It's the merchant transaction id passed to Amazon Pay. Maximum: 50 characters Recommended: 15 characters
referenceId	String	A unique id that can be used to tag multiple transactions together. Could be same as chargeId.
merchantId	String	Your merchant ID with Amazon Pay.
attributableProgram	String	You have to pass it as ***S2SPay***
noteToCustomer	String	***Optional***. Merchant notes to the customers. This will also appear in the settlement report. Maximum: 255 characters
customData	String	***Optional.*** Additional value for self-use. This field is not stored by Amazon but will be returned back in the callback. Maximum: 255 characters
timeoutInSecs	String	***Optional.*** To set a transaction timeout value in seconds. Amazon's default timeout is 15 min and the minimum value is 5 min.

Sample JSON for request:

```
{
  "intent": "Capture",
  "merchantId": "A28RUGPVUTQXU1",
  "chargeId": "ve7v676dsdsde7f",
  "accessToken": "A3RYF1HD1W5QCF",
  "referenceId": "ve7vdsdsde7f",
  "amount": "0.01",
  "currencyCode": "INR",
  "attributableProgram": "S2SPay ",
  "callbackUrl": "http://merchantx.com/pay",
  "noteToCustomer": "note to customer",
  "customData": "data",
  "timeoutInSecs": "90000"
}
```

Response Parameters:

response:

Parameter	Type	Description
merchantId	String	Your merchant ID with Amazon Pay.
requestedAmount	String	Amount requested as part of the request.
approvedAmount	String	Approved amount from Amazon Pay.
amazonChargeId	String	Amazon generated unique transaction Id for a given transaction.

chargeId	String	Merchant transaction id sent as part of the request.
status	String	This will indicate the status of the request. Possible values are – CapturePending /CaptureApproved *For Failure:* Declined
currencyCode	String	Will always be ***INR***
amazonPayUrl	String	***Optional***. This will be sent only in case the intent as *Capture* and customer does to have sufficient balance in APB wallet.
customData	String	***Optional.*** Will be part of the response only if the parameter was passed in the request
createTime	String	When the request was created at our end.
updateTime	String	When the request was last updated and sent back.

Sample Response:


```
{
  "merchantId": "A28RUGPVUTQXU1",
  "chargeId": "ve7vdsd99sde7f",
  "amazonChargeId": "P04-0249917-0735882",
  "requestedAmount": "0.01",
  "approvedAmount": "0.01",
  "currencyCode": "INR",
  "status": "CaptureApproved",
  "amazonPayUrl": "https://amazonpay.amazon.in/~",
  "customData": "data passed in the request"
  "createTime": "2020-01-13T05:28:30.377Z",
  "updateTime": "2020-01-13T05:28:30.377Z"
}
```


TopUp:

This API will help with top-up of Amazon Pay balance (ABP) for a user on your platform. Once you call this API, you will receive a top-up link to which you need to redirect the users so that they can top-up up their ABP. ReferenceId passed in this API should be used in the subsequent block/charge or capture request.

Method: POST
Path: <endpoint>/v1/payments/topup
ContentType: application/json

Request Parameters:

Body Fields:

Parameter	Type	Description
amount	String	Contains the amount to be charged for the order. You can only send up to 2 digits after decimal point.
currencyCode	Sting	Will always be ***INR***
callbackUrl	String	URL where the user will be redirected back post payment completion in the case of multi-tender payment. We need this for **whitelisting** at our end and the URL should be a static one. For transaction via Apps the URL has to be specific to our requirements that can be found under Appendix section.
accessToken	String	This is the access_token generated for the user. It will be used to identify the user at our end.
chargeId	String	Unique id generated by merchant for every transaction. It's the merchant transaction id passed to Amazon Pay. Maximum: 50 characters Recommended: 15 characters
referenceId	String	A unique id that can be used to tag multiple transactions together. In case of top-up request, do ensure you pass this value same for the next block and capture requests.
merchantId	String	Your merchant ID with Amazon Pay.
attributableProgram	String	You have to pass it as ***S2SPay***
noteToCustomer	String	***Optional***. Merchant notes to the customers. This will also appear in the settlement report.
customData	String	***Optional.*** Additional value for self-use. This field is not stored by Amazon but will be returned back in the callback.
timeoutInSecs	String	***Optional.*** To set a transaction timeout value in seconds. Amazon's default timeout is 15 min

Sample JSON for request:

```
{
  "merchantId": "A28RUGPVUTQXU1",
  "chargeId": "ve7v676dsdsde7f",
  "accessToken": "A3RYF1HD1W5QCF",
  "referenceId": "ve7vdsdsde7f",
  "amount": "0.01",
  "currencyCode": "INR",
  "attributableProgram": "S2SPay ",
  "callbackUrl": "http://merchantx.com/pay",
  "noteToCustomer": "note to customer",
  "customData": "data",  "timeoutInSecs": "90000"}
```

Response Parameters:

response:

Parameter	Type	Description
merchantId	String	Your merchant ID with Amazon Pay.
amount	String	Amount requested as part of the request.
amazonChargeId	String	Amazon generated unique transaction Id for a given transaction.
chargeId	String	Merchant transaction id sent as part of the request.
currencyCode	String	Will always be ***INR***
amazonPayUrl	String	Users have to be redirected to this URL so that they can top up their Amazon Pay balance.
customData	String	***Optional.*** Will be part of the response only if the parameter was passed in the request
createTime	String	When the request was created at our end.
updateTime	String	When the request was last updated and sent back.

Sample Response:


```
{
  "merchantId": "A2UOYLUXLC6NR3",
  "chargeId": "ve7v676dsdsde7f",
  "amazonChargeId": "P04-7502344-4500241",
  "amazonPayUrl": "https://amazonpay.amazon.in/payments/topup?...."
  "amount": "0.01",
  "currencyCode": "INR",
  "customData": "data",
  "createTime": "2020-01-18T11:11:41.511Z",
  "updateTime": "2020-01-18T11:12:52.661Z"
}
```


Refund:

This API allows you to refund a transaction. Refund is processed depending on the value that is being passed in the request

Method: POST
Path: <endpoint>/v1/payments/refund
ContentType: application/json

Request Parameters:

Body Fields:

Parameter	Type	Description
amount	String	Contains the amount to be charged for the order. You can only send up to 2 digits after decimal point.
currencyCode	Sting	Will always be ***INR***
refundId	String	Merchant generated unique identifier for each refund request. Use only the following characters: • lowercase a-z • uppercase A-Z • numbers 0-9 • dash (-) • underscore (_) Maximum: 32 characters
chargeIdType	String	The possible values are ***AmazonTxnId*** and ***MerchantTxnId***
chargeId	String	Appropriate id based on the type passed in the above field.
merchantId	String	Your merchant ID with Amazon Pay.
noteToCustomer	String	***Optional***. Merchant notes to the customers. This will also appear in the settlement report. Maximum: 255 characters
softDescriptor	String	Reason of refund. Maximum: 16 characters

Sample Request JSON:

```
{
  "merchantId": "A2UOYLUXLC6NR3",
  "refundId": "M04-3581290-6586191",
  "chargeIdType": "AmazonTxnId",
  "chargeId": "P04-0249917-0735882",
  "amount": "2.00",
  "currencyCode": "INR",
  "noteToCustomer": "string",
  "softDescriptor": "string"
}
```

Response Parameters:

response:

Parameter	Type	Description
refundId	String	A unique id generated for the refund request.
amazonRefundId	String	The Amazon-generated identifier for this refund transaction.

amount	String	The amount requested to be refunded.
refundedFee	String	The capture fee that has been refunded
currencyCode	Sting	Will always be ***INR***
status	String	Transaction status. Possible values are ***Approved, Pending, Declined***
createTime	String	When the request was created at our end.
updateTime	String	When the request was last updated and sent back.

Sample Response:


```
{
  "refundId": "ve7vdsd99sde7f",
  "amazonRefundId": "P04-0249917-0735882-R8908908",
  "amount": 0.01,
  "refundedFee": 0.01,
  "currencyCode": "INR",
  "status": "Approved/Pending/Declined",
  "createTime": "2020-01-18T11:11:41.511Z",
  "updateTime": "2020-01-18T11:12:52.661Z"
}
```


Status:

This API can be used to determine the status of either a charge or a refund transaction by passing the appropriate order id and type. You can pass either your transaction id or the amazon transaction id.

Polling strategy:

We recommend implementing the following polling strategy for status check:

```
1) 12 polls at 5 sec
2) 18 polls at 10 sec
3) Poll every 60 seconds post that till timeout.
4) Move to a DLQ/fail it if non-terminal response after timeout and contact your integration POC at a
mazon
```

Method: GET
Path: <endpoint>/v1/payments/{txnType}

Possible values of txn_type:

Value	Description
charge	Pass this for charge status check.
refund	Pass this for refund status check.
topup	Pass this for topup status check.

Request Parameters:

Query Parameters:

Parameter	Type	Description
merchantId	String	Your merchant id with Amazon Pay
txnIdType	String	The possible values are ***AmazonTxnId*** ,***MerchantTxnId***
txnId	String	**For Charge & Topup:** Pass the appropriate captureId depending on the type. **For Refund:** You can pass one of the below values as per your requirements **amazonRefundId**, **refundId.**

Possible values of txnId:

Parameter	Description
amazonRefundId	amazonRefundId value is returned on calling refund API. Formet : P04-XXXXXXX- XXXXXXX -R XXXXXXX
refundId	It is the unique identifier value you pass while calling the refund API.

Request Sample for charge:

https://{{endpoint}}/v1/payments/charge?merchantId=A2UOYLUXLC6NR3&txnIdType=AmazonTxnId&txnId=P04-4734274-8283393

Request Sample for Refund:

Request sample for amazonRefundId

https://{{endpoint}}/v1/payments/charge?merchantId=A2UOYLUXLC6NR3&txnIdType=AmazonTxnId&txnId=P04-0572904-1255078-R073442

Request sample for Merchant refundId

https://{{endpoint}}/v1/payments/charge?merchantId=A2UOYLUXLC6NR3&txnIdType=MerchantTxnId&txnId=TEST-7529519-0685907-R4

Response Parameters:

Parameter	Type	Description

amount	String	Amount of the transaction id passed or the refund amount.
currencyCode	String	Will always be ***INR***
amazonChargeId	String	Amazon generated unique order Id for a transaction.
chargeId	String	Unique id sent in the charge request.
status	String	Status of the request. The values depend on the type of status check. The possible values are: ***For Charge:*** CapturePending/CaptureApproved/Declined ***For refund:*** Approved/Pending/Declined
requestedAmount	String	Amount requested as part of the request. (only sent for charge status response)
approvedAmount	String	Approved amount from Amazon Pay. (only sent for charge status response)
createTime	String	When the request was created at our end.
updateTime	String	When the request was last updated and sent back.
amazonRefundId	String	This field is part of the response only for refund status check. (Only sent for refund status response)
refundId	String	The refund id shared for the refund request. (Only sent for refund status response)
refundedFee	String	The fee refunded (Only sent for refund status response)

Sample Response for Charge:


```
{
  "merchantId": "A2UOYLUXLC6NR3",
  "chargeId": "ve7v676dsdsde7f",
  "amazonChargeId": "P04-7502344-4500241",
  "status": "CapturePending/CaptureApproved/Failed",
  "requestedAmount": "0.01",
  "approvedAmount": "0.01",
  "currencyCode": "INR",
  "customData": "data",
  "createTime": "2020-01-18T11:11:41.511Z",
  "updateTime": "2020-01-18T11:12:52.661Z"
}
```


Sample Response for top-up:

```
{
  "merchantId": "A2UOYLUXLC6NR3",
  "chargeId": "ve7v676dsdsde7f",
  "amazonChargeId": "P04-7502344-4500241",
  "status": "Pending/Approved/Failed",
  "amount": "0.01",
  "currencyCode": "INR",
  "customData": "data",
  "createTime": "2020-01-18T11:11:41.511Z",
  "updateTime": "2020-01-18T11:12:52.661Z"
}
```


Sample Response for Refund:


```
{
  "refundId": "ve7vdsd99sde7f",
  "amazonRefundId": "P04-0249917-0735882-R8908908",
  "amount": 0.01,
  "refundedFee": 0.01,
  "currencyCode": "INR",
  "status": "Approved/Pending/Declined",
  "createTime": "2020-01-18T11:11:41.511Z",
  "updateTime": "2020-01-18T11:12:52.661Z"
}
```


Sample Response for Refund Details


```
// amazonRefundId / refundId
{
“refundId” : “TEST-7529519-0685907-R4”,
"amazonRefundId" : "P04-XXXXXXX- XXXXXXX - XXXXXXX ",
"amount" : "4.00",
"refundedFee" : "0.00",
"currencyCode" : "INR",
"status" : "Approved",
"createTime" : "2023-02-16T05:30:40.689Z",
"updateTime" : "2023-02-16T05:30:53.322Z"
}
```


Signature Generation and Verification

In order to make an API request you need to pass a signature as part of the request. Also, you should be validating the signature returned in the API response to make sure that the response is originating from Amazon. To generate/verify a signature ***HMAC-SHA-384*** algorithm and Amazon Signature v4 process will be used.

Steps to generate the signature:

1. Create a canonical request. (Here is a list of sample canonical request for every APIs)
2. Use the canonical request and additional metadata to create a string for signing.

3. Generate a signature key using the secret key provided by Amazon.
4. Then use the signature key from step 3 and the string from step 2 to create a signature.

For detailed information on signature generate please follow this guide - https://amazonpay.s3-us-west-2.amazonaws.com/Signature/RequestswithSignatureVersion6.pdf

Amazon will provide a 384bit secret key. Each key is identified with a public *access key Id* which is to be passed in the HTTP **Authorization** header. Each key has an expiry time associated with it. Amazon will not accept request signed with expired keys or will never sign responses with such keys.

The list of all the secret keys will be available at https://amazonpay.amazon.in/integration-details

Expiry time of the key:
This will be used at the time of key rotation (to invalidate the old key). The key shared initially and the new key after rotation won't have an expiry time associated, meaning we are not going to expire it in near future.

Key rotation:
Secret keys will be rotated at regular intervals for two reasons:

1. **To meet security compliances**: In such cases you will be notified of this key rotation beforehand. Once the key is rotated you will see two keys on the portal for a certain period of time. The old key will now have an expiry period before which you must start using the new key.

2. **To mitigate the effect of compromised keys**: In case of key being compromised we will rotate the keys immediately and invalidate the compromised key.

Amazon will verify the request and sign the response for the request using the same keyId that is passed in the request. This should be taken care of especially in case when more than one valid key exists (rotation period).

NOTE: These are secret(private) keys and only Amazon and the merchant should have access to these keys. These keys should never be stored as plaintext anywhere.

Testing:

- Test account linking and delinking feature and handle both scenarios.
- Test a sufficient balance transaction as well as an insufficient balance transaction.
- Test balance fetch before transaction and after transaction.

Things to remember:

- The status validation is a must for confirming a transaction.

- Signature verification must be done for all the API responses.
- User tokens should always be stored in an encrypted format into your database.

Appendix

Generic error codes for APIs:

HTTP status code	Code	Message
400	InvalidArguments	The value of an argument is invalid.
	InvalidMerchantId	Specified merchant id does not exist.
	InvalidChargeId	Specified charge id does not exist.
	InvalidRefundId	Specified refund id does not exist.
	InvalidCallbackUrl	Specified callback url is not whitelisted.
	InactiveAmazonPayBalanceAccount	Specified Amazon Pay Money balance account is not active.
	NonKYCAccount	Specified Amazon Pay Money balance account is not KYC compliant.
	InsufficientFunds	Charge could not be processed due to insufficient funds.
	InsufficientLoadLimit	Charge could not be processed due to insufficient load limit.
	DuplicateChargeId	Specified charge id has already been processed.
	DuplicateRefundId	Specified refund id has already been processed.
	ExcessRefundAmount	Specified refund amount exceeds allowable limit.
	ChargeNotApproved	Specified charge id is not approved.
403	InvalidAccessToken	Merchant is not authorized. The access token may be either malformed or unusable.
401	AuthorizationExpired	Specified signature has expired.
	AuthorizationFailed	Specified signature is invalid.
500	InternalServerError	An internal server error has occurred. Try your request again later.

Error Sample:

```
{
    "code": "InvalidMerchantId",
    "message": "Specified merchant id does not exist"
}
```

Amazon Pay Logo:
There are two logos (a) Long form and (b) Short form.

- Long form – To be used on merchant banners around specific promotions
- Short form - To be used instead of old wallet icon logo. Typically, this was used on the merchant PSP only

Here is a downloadable link to download the log and guidelines - https://amazonpay.s3-us-west-2.amazonaws.com/logos/amazon_pay_logo_pack_Guidelines._V535346670_.zip

callbackURL for Apps:
These URLs has to be specific to the below pattern for the SDK to read it and close the chrome custom tab. Also, do reach out to your integration specialist at Amazon Pay to get these whitelisted.

Android: amzn://amazonpay.amazon.in/<your package name>
iOS: amzn-<bundle identifier of the app>://pwa