

Transaction States and Errors

[Suggest Edits](#)

API entities, such as transactions and challenges, are processed asynchronously after the initial POST request.

As the request is processed, the transaction's state, or the challenge's status, will update accordingly. This update can be retrieved either by polling the associated GET endpoint or via [subscribing to notifications](#) (recommended).

Transaction States

Transactions go through various states after initiation to indicate if they've begun processing, are identifiable on the blockchain, complete, failed, and more.

Specific actions, like canceling or accelerating a transaction, can only be taken when a transaction is in a specific state.

Canceling a transaction can only occur when a transaction is in the INITIATED , QUEUED , or SENT states.

Accelerating a transaction can only occur when a transaction is in the SENT state.

If a transaction cannot be successfully completed, it will enter the FAILED state and have an accompanying `errorReason` detailing the failure.

The following state diagram shows the flow for transaction states:

The following table describes the possible states of a transaction:

State	Description
INITIATED	The transaction has been initiated.
QUEUED	The transaction is in the processing queue.
PENDING_RISK_SCREENING	The transaction is waiting for risk screening.
SENT	The transaction has been identified in the Mempool and assigned a transaction hash. The transaction may be accelerated at this stage.
CONFIRMED	The transaction has been identified on a mined block. The transaction can no longer be canceled.
COMPLETED	Terminal State. The transaction has successfully completed.
CANCELED	Terminal State. The transaction has been canceled and must be re-initiated.
FAILED	Terminal State. The transaction has failed. The transaction must be re-initiated.
DENIED	Terminal State. The platform denies the transaction. For a detailed reason, refer to the <code>errorReason</code> .
ACCELERATED	The transaction has been accelerated.

Transaction Errors

If a transaction cannot be completed and enters the FAILED state, the transaction object will have a related `errorReason` to describe the failure.

Error Reason	Description
ESTIMATION_ERROR	The transaction estimation failed during node execution.
INSUFFICIENT_NATIVE_TOKEN	The wallet does not have a sufficient balance of the blockchain's native asset to cover the gas fees needed for all pending transactions.
ABI_SIGNATURE_PARAMS_MISMATCH	The signature and parameter of ABI are mismatched.
FEE_EXCEEDS_MAX_ALLOWANCE	The fee exceeds the max allowance. There is one native token fee limit to prevent any unintended transaction from being sent out.
QUERY_NFT_METADATA_ERROR	The NFT metadata cannot be found.
GAS_LIMIT_TOO_LOW	The provided fee tolerance (<code>gasLimit</code>) was too low to complete the requested transaction.
TRANSACTION_UNDER_PRICE	The transaction's max fee or priority fee is lower than the current lowest max fee or priority fee in mempool when mempool is full.
FAILED_ON_CHAIN	The transaction failed to be processed by the blockchain.
FAILED_REORG	The transaction was dropped because of blockchain re-org.
INTERNAL_ERROR	The transaction failed due to an internal error.
PAYMASTER_POLICY_NOT_ACTIVATED	The paymaster policy hasn't been set or activated.
PAYMASTER_POLICY_EXCEED_MAX_DAILY_TRANSACTIONS	The maximum number of daily transactions allowed by the paymaster policy has been reached. Please try again tomorrow or update the policy limits.
PAYMASTER_POLICY_EXCEED_MAX_SPEND_USD_PER_TX	The transaction exceeds the maximum allowable spend in USD as per the paymaster policy for a single transaction. Please increase the limits in your policy for such transactions to go through.
PAYMASTER_POLICY_EXCEED_MAX_SPEND_USD_DAILY	The maximum daily spending limit in USD set by the paymaster policy has been reached. Please try again tomorrow or update the policy limits.
PAYMASTER_POLICY_EXCEED_MAX_NATIVE_TOKEN_DAILY	The maximum daily spending limit for native tokens set by the paymaster policy has been reached. Please try again tomorrow or update the policy limits.
PAYMASTER_POLICY_SENDER_IN_BLOCKLIST	The sender wallet of the transaction is blocked by the paymaster policy. Update the policy blocklist to allow this user to go through.
PAYMASTER_ENTITY_RESTRICTION	At this time, your account is restricted from using Gas Station. Go to the Web3 Services Console Paymaster page to learn more.

Challenge States

Challenges go through several statuses to indicate if it has been presented to the user, have been completed, and more.

If a challenge cannot be successfully completed, it will enter the FAILED state, and have an accompanying `errorCode` and `errorMessage` detailing the failure.

Status Description PENDING The request has been created but has not yet been approved by the user. IN_PROGRESS The user has approved the request, and it is being executed. COMPLETED The request is complete. FAILED The request has failed to execute. The request must be re-initiated. EXPIRED The request has expired. The request must be re-initiated.

Updated 1 day ago * [Table of Contents](#) * * [Transaction States](#) * * [Transaction Errors](#) * * [Challenge States](#)