Common Errors

We are sorry to cause any inconvenience while interacting with biconomy SDK. We are trying our best to send meaningful error messages that can help you do quick fixes without interacting with any support channel. Some of the error messages are not in our control as we are using some standard contract (e.g EntryPoint) to provider gasless experience. Following are some standard error messages that can help you apply quick fixes and sort the issues.

tip Ensure that you are using the bundler, paymaster, and RPC URL of the same blockchain. Error Description AA21 didn't pay prefund Throws if your smart wallet does not have funds to send transaction. Send some native tokens in your smart wallet to be able to resolve the error. AA23 reverted (or OOG) Make sure the signature verification logic of the smart account is implemented correctly. Ensure you are using the same signer, you created the account with. If the error persists, try simulatingvalidateUserOp method on entrypoint contract with the user operation ontenderly . Also ensure, if sender has sufficient native tokens to cover the User Operation's gas costs. If you are using paymaster to sponsor the transactions, ensure that the paymasterAndData field of the user operation is correctly set to enable proper handling of gas fees. AA24 Signature error Throws if the signature field of userOp is inaccurate. This could occur if any userOp field is altered after the userOp has been signed. Ensure that you are using the correct chain Id, entry point address and corresponding bundler URL. AA10 sender already constructed Throws if your smart wallet is already created but you are still sendinginitcode in userOp. AA13 initCode failed or OOG Throws if userOp doesn't have enoughverificationGasLimit to create smart account. you need to increaseverificationGasLimit to be able to send transaction. AA14 initCode must return sender Throws if the factory contract that you are using to deploy smart account does not return smart account address. Orsender field in userOp does not have same address as factory contract is creating for you. AA93 invalid paymasterAndData Throws ifpaymasterAndData created by signing service is not valid. AA95 out of gas Throws ifcallGasLimit orverificationGasLimit is not correct for executing callData sent in userOp. Make surecallGasLimit andverificationGasLimit are not too high and use correct estimated values for preparing the userOp. AA90 invalid beneficiary Throws if the beneficiary address sent in userOp that is going to get sponsored transaction fee back is not valid. AA31 paymaster deposit too low Throws if the Paymaster you have created using dashboard does not have enough funds in its gas tank to sponsor transaction. AA41 too little verificationGas Throws if the userOp does not have enoughverificationGasLimit to create smart account. you need to increaseverificationGasLimit to be able to resolve this error. AA33 reverted (or OOG) Throws if the transaction you are sending is not valid or userOp does not have enoughverificationGasLimit to be able to validate transaction. Try increasingverificationGasLimit or verify either the transaction you are making is valid to be able to resolve this error. In case of SPONSORED paymaster you might get this error due to insufficient token balance. AA40 over verificationGasLimit Throws when gas used in the verification step on chain, is more than theverificationGasLimit . Try increasing theverificationGasLimit value to be able to resolve this error. AA51 prefund below actualGasCost Throws because of 2 possible reasons: either your smart wallet does not have funds to send transaction or the Paymaster you have created using dashboard does not have enough funds in its gas tank to sponsor transaction. AA34 signature error Throws when the paymaster's signature is invalid. Check the format of the signature in paymasterAndData. Once the paymasterData is created, any changes to the userOp may lead to this error. In case you have passed thecalculateGasLimits flag as true while creating the paymasterData, do update the gasLimits as mentionedhere to avoid this error. AA25 Invalid Account Nonce Throws when the nonce is invalid. The user operation may be re-using an old nonce, or formatted the nonce incorrectly. If you are trying parallel transactions execution, refer to nonceOptions inthis page for more details No policies were set on the dashboard. Please set policies to allow gas sponsorship via paymasters Throws if the contracts are not whitelisted using the dashboard to be able to sponsor transactions. error in txn Throws when there is internal server error from relayer side which can be due to multiple reasons. In this we would require the transacitonId from the user. only allowed via delegateCall Throws when trying to send native tokens directly to the Smart Account implementation. Smart Account:: new Signatory address cannot be self Throws when trying to set the Smart Account address as its own owner. new Signatory address cannot be same as old one Throws when trying to set owner with an address that is already an owner. Address cannot be zero Throws when trying to update to a zero address as implementation. invalid tokenGasPriceFactor Throws when token Gas Price factor provided is 0. Invalid signatures length Throws when the length of the signature provided in the Forward flow is less than 65. Could not get network config values Throws when we can't find the provided network in the network config. Exceeded maximum duration (30 sec) waiting to get receipt for userOpHash This is a timeout error on bundler, If the error persists on a particular network, create a bundler instance by increasing the userOpReceiptMaxDurationIntervals as mentionedhere . And pass this bundler to thecreateSmartAccountClient function UserOperation reverted during execution phase This error comes when the call data execution is failing, which is the call that the Entry Point makes to the Smart Account. A quick way to debug is to go on Tenderly and simulate the exact call data execution. To do that you can add the Smart Account address in thePaste any address box of Tenderly, then select the network which you are on. After that paste thecallData from the userOp in the raw data input box and also edit the From address to be the Entry Point address. If the network you are on is not supported by Tenderly, one can useeth_estimateGas to capture the revert reason.

Custom Errors

Error Signature Error Selector Description CallerIsNotAnEntryPoint (address) 0xbb587b6e Throws at onlyEntryPoint when msg.sender is not an EntryPoint set for this Smart Account HandlerCannotBeZero() 0xdd449f5f Throws if zero address has been provided as Fallback Handler address EntryPointCannotBeZero() 0x245d23e4 Throws if zero address has been provided as Entry Point address MixedAuthFail (address) 0x1141614a Throws at mixedAuth when msg.sender is not an owner neither _self TokenTransferFailed (address,address,uint256) 0xc8776798 Throws if transfer of tokens failed. Arguments: token address, receiver address, amount. OwnerCannotBeZero() 0x9b15e16f Throws if trying to change an owner of a SmartAccount to the zero address BaseImplementationCannotBeZero() 0x70204800 Throws if zero address has been provided as Base Implementation address InvalidImplementation (address) 0x0c760937 Throws if there is no code at implementationAddress CallerIsNotOwner (address) 0xd4ed9a17 Throws at onlyOwner when msg.sender is not an owner CallerIsNotEntryPointOrOwner (address) 0x65b7a78e Throws at _requireFromEntryPointOrOwner when msg.sender is not an EntryPoint neither an owner AlreadyInitialized (address) 0x93360fbf Throws if trying to initialize a Smart Account that has already been initialized NotEnoughGasLeft (uint256,uint256) 0xbbbb17a0 Throws if not enough gas is left at some point CanNotEstimateGas (uint256,uint256,bool) 0x830fc3f8 Throws if not able to estimate gas. It can happen when the amount of gas and its price are both zero and the transaction has failed to be executed WrongContractSignatureFormat (uint256,uint256,uint256) 0x71448bfe Throws if contract signature is provided in the wrong format WrongContractSignature (bytes) 0x605d3489 Throws when isValidSignature for the contract signature and data hash return differs from EIP1271 Magic Value InvalidSignature (address,address) 0x42d750dc Throws when the address that signed the data (restored from signature) differs from the expected signer ExecutionFailed() 0xacfdb444 Throws when the transaction execution fails TransferToZeroAddressAttempt() 0x9293b190 Throws when trying to transfer to the zero address WrongBatchProvided (uint256,uint256,uint256) 0x50605488 Throws when data for executeBatchCall provided in the wrong format (i.e. empty array or lengths mismatch) ProxyDeploymentFailed (address,uint256) 0x6d05b867 Throws when the Proxy (Smart Account) deployment attempt failed ModulesAlreadyInitialized() 0xdf8cc4e3 Throws when trying to initialize the module manager that has already been initialized ModulesSetupExecutionFailed() 0x65c74720 Throws when a delegatecall during module manager initialization has failed ModuleCannotBeZeroOrSentinel (address) 0xcadb248f Throws when address(0) or SENTINEL_MODULES constant has been provided as a module address ModuleAlreadyEnabled (address) 0xb29d4595 Throws when trying to enable a module that has already been enabled ModuleAndPrevModuleMismatch (address,address,address) 0xc40d496c Throws when the module and previous module mismatch ModuleNotEnabled (address) 0x21ac7c5f Throws when trying to execute a transaction from a module that is not enabled CallerIsNotSelf (address) 0x051e38cc Throws when the caller is not address(this) EntryPointCannotBeZero() 0x245d23e4 Throws when the Entry Point address provided is address(0) VerifyingSignerCannotBeZero() 0x8fc6a931 Throws when the verifying signer address provided is address(0) PaymasterIdCannotBeZero() 0xab9a38ca Throws when the paymaster Id address provided is address(0) DepositCanNotBeZero() 0x674c2ee2 Throws when 0 has been provided as the deposit amount CanNotWithdrawToZeroAddress() 0x92bc9df3 Throws when trying to withdraw to address(0) InsufficientBalance (uint256,uint256) 0xcf479181 Throws when trying to withdraw more than the available balance InvalidPaymasterSignatureLength (uint256) 0xe4b52b17 Throws when the signature provided to the paymaster has an invalid length

Previous Troubleshooting Next Polyfill Errors