Errors API Reference

Add Chainlink CCIP to your project

If you need to integrate Chainlink CCIP into your project, install the@chainlink/contracts-ccip NPM package .

npmyarnIf you useNPM :

npminstall@chainlink/contracts-ccip--saveIf you useYarn :

yarnadd@chainlink/contracts-ccip

When invoking theccipSendfunction , it is possible to encounter various errors. These might be thrown either by the CCIP router or by one of the downstream contracts called by the CCIP router. Below is a compiled list of potential errors you might encounter. Referencing this list will enable you to capture and handle these exceptions gracefully.

Router

ErrorParametersDescriptionUnsupportedDestinationChain* uint64 destChainSelectorThrown when the destination chain is not supported.InsufficientFeeTokenAmount-Thrown when the CCIP fees are paid with native tokens, but not enough is sent with the transaction.InvalidMsgValue-Thrown when the CCIP fees arenotpaid in native tokens, butmsg.valueis non-zero.

Onramp

ErrorParametersDescriptionNotAFeeToken *address tokenThrown when an unsupported fee token is used.UnsupportedToken* IERC20 tokenThrown when an unsupported transfer token is used.InvalidAddress*bytes encodedAddressThrown when the receiver address is invalid.InvalidExtraArgsTag-Thrown when an invalid extra arguments tag is used.RouterMustSetOriginalSender-This error should never be thrown as the router always sets the sender.MustBeCalledByRouter-This error should never be thrown as the router always makes the call.MessageGasLimitTooHigh-Thrown when the gas limit is too high.UnsupportedNumberOfTokens-Thrown when too many tokens are involved in the transfer.SenderNotAllowed* address senderThrown when the sender is not allowlisted.MaxFeeBalanceReached-Thrown when the onRamp has reached its maximum fee storage capacity. If it is full, it cannot process new transactions.

RateLimiter

ErrorParametersDescriptionPriceNotFoundForToken *address tokenThrown when a price cannot be found for a specific token.BucketOverfilled-This error should never be thrown as it indicates an invalid bucket state.AggregateValueMaxCapacityExceeded* uint256 capacity * uint256 requestedThrown when the user requests to transfer more value than the capacity of the aggregate rate limit bucket.TokenMaxCapacityExceeded *uint256 capacity * uint256 requested * address tokenAddressThrown when the user requests to transfer more of a token than the capacity of the bucket.AggregateValueRateLimitReached* uint256 minWaitInSeconds * uint256 availableThrown when the user requests to transfer more value than currently available in the bucket. The user might have to wait for at leastminWaitInSecondsfor enough availability or transfer the currentlyavailableamount.TokenRateLimitReached* uint256 minWaitInSeconds * uint256 available * address tokenAddressThrown when the user requests to transfer more of a token than currently available in the bucket. The user might have to wait at leastminWaitInSecondsfor enough availability, or transfer the currentlyavailableamount.

ERC20

ErrorDescriptionERC20: burn amount exceeds balanceThrown when the amount to be burned exceeds the pool balance.ERC20: transfer amount exceeds allowanceThrown when the transfer amount exceeds the allowance.

PriceRegistry

ErrorParametersDescriptionChainNotSupported *uint64 chainThrown when a chain is not supported.StaleGasPrice* uint64 destChainSelector * uint256 threshold * uint256 timePassedThrown when the gas price is stale.TokenNotSupported *address tokenThrown when a token is not supported.StaleTokenPrice* address token * uint256 threshold * uint256 timePassedThrown when the price of a token is stale.