

Errors API Reference

Integrate Chainlink CCIP into your project

npm yarn foundry If you use [NPM](#), install the [@chainlink/contracts-ccip NPM package](#) and set it to the v1.4.0 release:

npm install @chainlink/ If you use [Yarn](#), install the [@chainlink/contracts-ccip NPM package](#) and set it to the v1.4.0 release:

yarn add @chainlink/ If you use [Foundry](#), install the v1.4.0 release:

forge install smartcontractkit/ccip@b06a3c2e6cb9892ec6f76a015624413ffa1a122

When invoking the [ccipSend function](#), 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

ErrorParametersDescription **UnsupportedDestinationChain** * uint64 destChainSelector Thrown 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 are not paid in native tokens, but msg.value is non-zero.

Onramp

ErrorParametersDescription **NotAFeeToken** address token Thrown when an unsupported fee token is used. **UnsupportedToken** IERC20 token Thrown when an unsupported transfer token is used. **InvalidAddressBytes** encodedAddress Thrown 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 sender Thrown when the sender is not allowed. **MaxFeeBalanceReached** - Thrown when the onRamp has reached its maximum fee storage capacity. If it is full, it cannot process new transactions.

RateLimiter

ErrorParametersDescription **PriceNotFoundForToken** address token Thrown 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 requested Thrown when the user requests to transfer more value than the capacity of the aggregate rate limit bucket. **TokenMaxCapacityExceeded** uint256 capacity * uint256 requested * address token Address Thrown when the user requests to transfer more of a token than the capacity of the bucket. **AggregateValueRateLimitReached** uint256 minWaitInSeconds * uint256 available Thrown when the user requests to transfer more value than currently available in the bucket. The user might have to wait for at least minWaitInSeconds for enough availability or transfer the currently available amount. **TokenRateLimitReached** * uint256 minWaitInSeconds * uint256 available * address token Address Thrown when the user requests to transfer more of a token than currently available in the bucket. The user might have to wait at least minWaitInSeconds for enough availability, or transfer the currently available amount.

ERC20

ErrorDescription **ERC20: burn amount exceeds balance** Thrown when the amount to be burned exceeds the pool balance. **ERC20: transfer amount exceeds allowance** Thrown when the transfer amount exceeds the allowance.

PriceRegistry

ErrorParametersDescription **ChainNotSupported** uint64 chain Thrown when a chain is not supported. **StaleGasPrice** uint64 destChainSelector * uint256 threshold * uint256 timePassed Thrown when the gas price is stale. **TokenNotSupported** address token Thrown when a token is not supported. **StaleTokenPrice** address token * uint256 threshold * uint256 timePassed Thrown when the price of a token is stale.