Client Library 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

CCIP senders and receivers use theCCIP Client Libraryto build CCIP messages.

import{Client}from"@chainlink/contracts-ccip/src/v0.8/ccip/libraries/Client.sol";

Types and Constants

EVMTokenAmount

Use this solidity struct to specify the token address and amount.

structEVMTokenAmount{addresstoken;uint256amount;} NameTypeDescriptiontokenaddresstoken address on the local chain.amountuint256Amount of tokens.

Any2EVMMessage

CCIP receivers use this solidity struct to parse the received CCIP message.

structAny2EVMMessage{bytes32messageId;uint64sourceChainSelector;bytessender;bytesdata;structClient.EVMTokenAmount[]destTokenAmounts;} NameTypeDescriptionmessageIdbytes32CCIP messageId, generated on the source chain.sourceChainSelectoruint64Source chain selector.senderbytesSender address.abi.decode(sender, (address))if the source chain is an EVM chain.databytesPayload sent within the CCIP message.destTokenAmountsClient.EVMTokenAmount[]Tokens and their amounts in their destination chain representation.

EVM2AnyMessage

CCIP senders use this solidity struct to build the CCIP message.

structEVM2AnyMessage{bytesreceiver;bytesdata;structClient.EVMTokenAmount[]tokenAmounts;addressfeeToken;bytesextraArgs;} NameTypeDescriptionreceiverbytesReceiver address. Useabi.encode(sender)to encode the address tobytes.databytesPayload sent within the CCIP message.tokenAmountsClient.EVMTokenAmount[]Tokens and their amounts in the source chain representation.feeTokenaddressAddress of feeToken. Setaddress(0)to pay in native gas tokens such as ETH on Ethereum or MATIC on Polygon.extraArgsbytesUsers fill in theEVMExtraArgsV1 struct then encode it to bytes using the_argsToBytes function

EVM_EXTRA_ARGS_V1_TAG

bytes4EVM_EXTRA_ARGS_V1_TAG

EVMExtraArgsV1

structEVMExtraArgsV1{uint256gasLimit;} NameTypeDescriptiongasLimituint256specifies the maximum amount of gas CCIP can consume to executeccipReceive()on the contract located on the destination blockchain. ReadSetting gasLimit for more details.

Functions

_argsToBytes

function_argsToBytes(structClient.EVMExtraArgsV1 extraArgs)internalpurereturns(bytesbts) It is used to convert the arguments to bytes.

Parameters

NameTypeDescriptionextraArgsClient.EVMExtraArgsV1Extra arguments.

Return Values

NameTypeDescriptionbtsbytesEncoded extra arguments inbytes.