Interfaces

Mainnet Access

Chainlink Data Streams is available on Arbitrum Mainnet and Arbitrum Sepolia.

Talk to an expert

Contact us to talk to an expert about integrating Chainlink Data Streams with your applications.

Data Streams require several interfaces in order to retrieve and verify reports.

- Automation interfaces:* StreamsLookupCompatibleInterface
- ILogAutomation
- Data Streams interfaces:* IVerifierProxy
- IReportHandler

In the current code example for using Data Streams with Automation, these interfaces are specified in the example itself. Imports for these interfaces will be available in the future.

```
// SPDX-License-Identifier: MITpragmasolidity0.8.19;import{Common}from"@chainlink/contracts/src/v0.8/llo-
feeds/libraries/Common.sol";import{StreamsLookupCompatibleInterface}from"@chainlink/contracts/src/v0.8/automation/interfaces/StreamsLookupCompatibleInterface.sol";import{ILogAutomation,Log}fro
feeds/interfaces/IRewardManager.sol";import{IVerifierFeeManager}from"@chainlink/contracts/src/v0.8/llo-
feeds/interfaces/IVerifierFeeManager.sol";import{IERC20}from"@chainlink/contracts/src/v0.8/vendor/openzeppelin-solidity/v4.8.3/contracts/interfaces/IERC20.sol";/ * THIS IS AN EXAMPLE
CONTRACT THAT USES UN-AUDITED CODE FOR DEMONSTRATION PURPOSES. * DO NOT USE THIS CODE IN PRODUCTION. */// Custom interfaces for IVerifierProxy and
IFeeManagerinterfaceIVerifierProxy{functionverify(bytescalldatapayload,bytescalldataparameterPayload)externalpayablereturns(bytesmemoryverifierResponse);functions_feeManager()ext
The feed ID the report has data foruint32validFromTimestamp;// Earliest timestamp for which price is applicableuint32observationsTimestamp;// Latest timestamp for which price is
applicableuint192nativeFee;// Base cost to validate a transaction using the report, denominated in the chain's native token (WETH/ETH)uint192linkFee;// Base cost to validate a
transaction using the report, denominated in LINKuint32expiresAt;// Latest timestamp where the report can be verified onchainint192price;// DON consensus median price, carried to 8
decimal places}structPremiumReport{bytes32feedId;// The feed ID the report has data foruint32validFromTimestamp;// Earliest timestamp for which price is
applicableuint32observationsTimestamp;// Latest timestamp for which price is applicableuint192nativeFee;// Base cost to validate a transaction using the report, denominated in the
chain's native token (WETH/ETH)uint192linkFee;// Base cost to validate a transaction using the report, denominated in LINKuint32expiresAt;// Latest timestamp where the report can be
verified onchainint192price;// DON consensus median price, carried to 8 decimal placesint192bid;// Simulated price impact of a buy order up to the X% depth of liquidity
utilisationint192ask;// Simulated price impact of a sell order up to the X% depth of liquidity
utilisation}structQuote{addressquoteAddress;}eventPriceUpdate(int192indexedprice);IVerifierProxypublicverifier;addresspublicFEE_ADDRESS;stringpublicconstantDATASTREAMS_FEED
This example reads the ID for the basic ETH/USD price report on Arbitrum Sepolia.// Find a complete list of IDs at https://docs.chain.link/data-streams/stream-idsstring[]publicfeedIds=
["0x00027bbaff688c906a3e20a34fe951715d1018d262a5b66e38eda027a674cd1b"];constructor(address_verifier){verifier=IVerifierProxy(_verifier);}// This function uses revert to convey call
information.// See https://eips.ethereum.org/EIPS/eip-3668#rationale for
details.functioncheckLog(Logcalldatalog,bytesmemory)externalreturns(boolupkeepNeeded,bytesmemoryperformData)
{revertStreamsLookup(DATASTREAMS_FEEDLABEL,feedIds,DATASTREAMS_QUERYLABEL,log.timestamp,"");}/ * @notice this is a new, optional function in streams lookup. It is meant to
surface streams lookup errors. * @return upkeepNeeded boolean to indicate whether the keeper should call performUpkeep or not. * @return performData bytes that the keeper should call
performUpkeep with, if * upkeep is needed. If you would like to encode data to decode later, try abi.encode.
/functioncheckErrorHandler(uint256/errCode/,bytesmemory/extraData/)externalpurereturns(boolupkeepNeeded,bytesmemoryperformData){return(true,"0");// Hardcoded to always perform upkeep.//
Read the StreamsLookup error handler guide for more information.// https://docs.chain.link/chainlink-automation/guides/streams-lookup-error-handler}// The Data Streams report bytes is passed here.//
extraData is context data from feed lookup process.// Your contract may include logic to further process this data.// This method is intended only to be simulated offchain by Automation.// The data
returned will then be passed by Automation into performUpkeepfunctioncheckCallback(bytes[]calldatavalues,bytescalldataextraData)externalpurereturns(bool,bytesmemory)
{return(true,abi.encode(values,extraData));}// function will be performed onchainfunctionperformUpkeep(bytescalldataperformData)external{// Decode the performData bytes passed in by CL
Automation.// This contains the data returned by your implementation in checkCallback().(bytes[]memorysignedReports,bytesmemoryextraData)=abi.decode(performData,
(bytes[],bytes));bytesmemoryunverifiedReport=signedReports[0];(,/ bytes32[3] reportContextData */bytesmemoryreportData)=abi.decode(unverifiedReport,(bytes32[3],bytes));// Report verification
feesIFeeManager feeManager=IFeeManager(address(verifier.s_feeManager()));IRewardManager
rewardManager=IRewardManager(address(feeManager.i_rewardManager()));addressfeeTokenAddress=feeManager.i_linkAddress();
(Common.Assetmemoryfee,,)=feeManager.getFeeAndReward(address(this),reportData,feeTokenAddress);// Approve rewardManager to spend this contract's balance in
feesIERC20(feeTokenAddress).approve(address(rewardManager),fee.amount);// Verify the reportbytesmemoryverifiedReportData=verifier.verify(unverifiedReport,abi.encode(feeTokenAddress));//
Decode verified report data into BasicReport structBasicReportmemoryverifiedReport=abi.decode(verifiedReportData,(BasicReport));// Log price from reportemitPriceUpdate(verifiedReport.price);//
Store the price from the reportlast_retrieved_price=verifiedReport.price;}fallback()externalpayable{}}
```

Open in Remix What is Remix?