Feed Registry

The Chainlink Feed Registry is an onchain mapping of assets to feeds. It enables you to query Chainlink data feeds from asset addresses directly, without needing to know the feed contract addresses. They enable smart contracts to get the latest price of an asset in a single call, from a single contract.

Note

The Feed Registry only lists feeds with a canonical token address on a network. Non-token feeds such as stock indexes are not supported because they do not have a canonical token address.

For a complete list of functions and parameters for theFeedRegistryInterfacecontract, see theFeed Registry API Reference .

Base and Quote

The Feed Registry fully supports theAggregatorV3Interface API for multiple feeds. It maps feeds frombaseandquoteaddress pairs. To get the latest LINK / USD round data from the registry, call:

latestRoundData(addressbase,addressquote) For example, to get the latest LINK / USD price:

- base: The LINK token address on that network e.g.0x514910771AF9Ca656af840dff83E8264EcF986CAfor LINK on Ethereum mainnet
- quote: ADenominations.USDaddress (0x0000000000000000000000000000000000000348), which is based onISO 4217 .

latestRoundData(0x514910771AF9Ca656af840dff83E8264EcF986CA,0x0000000000000000000000000000000000000348) To get the latest LINK / ETH price on Ethereum:

- base: The LINK token address on that network e.g.0x514910771AF9Ca656af840dff83E8264EcF986CAfor LINK on Ethereum mainnet
- quote: ADenominations.ETHaddress (0xEeeeeEeeeEeEeeEeEeEeeEEEeeeeEeeeeeeeEEeE)

latestRoundData(0x514910771AF9Ca656af840dff83E8264EcF986CA,0xEeeeeEeeeEeEeeEeEeEeeEEEeeeeEeeeeeeeEEeE)

Denominations library

ADenominations Solidity library is available for you to fetch currency identifiers which lack a canonical Ethereum address:

// SPDX-License-Identifier: MITpragmasolidity^0.8.7;libraryDenominations{addresspublicconstantETH=0xEeeeeEeeeEeEeeEeEeEeeEEEeeeeEeeeeeeeEEeE;addresspublicconstantBTC=0xbBbBBBBbbBBBbbbBbbBbbbbBBbBbbbbBBbB... Fiat currencies follow https://en.wikipedia.org/wiki/ISO_4217addresspublicconstantUSD=address(840);addresspublicconstantGBP=address(826);addresspublicconstantEUR=address(978);// ... other fiat currencies} Open in Remix What is Remix?

Code Examples

Solidity

To consume price data from the Feed Registry, your smart contract should referenceFeedRegistryInterface , which defines the external functions implemented by the Feed Registry.

// SPDX-License-Identifier: MITpragmasolidity^0.8.7;import{FeedRegistryInterface}from"@chainlink/contracts/src/v0.8/interfaces/FeedRegistryInterface.sol";import{Denominations}from"@chainlink/contracts/src/v0.8/Denominations... *THIS IS AN EXAMPLE CONTRACT THAT USES HARDCODED VALUES FOR CLARITY. * THIS IS AN EXAMPLE CONTRACT THAT USES UN-AUDITED CODE. * DO NOT USE THIS CODE IN PRODUCTION. */contractPriceConsumer{FeedRegistryInterfaceinternalregistry;/* * Network: Ethereum Mainnet * Feed Registry: 0x47Fb2585D2C56Fe188D0E6ec628a38b74fCeeeDf /constructor(address_registry){registry=FeedRegistryInterface(_registry);}/ * Returns the ETH / USD price /functiongetEthUsdPrice()publicviewreturns(int){// prettier-ignore(/uint80 roundID/,intprice,/uint startedAt/,/uint timeStamp/,/uint80 answeredInRound/)=registry.latestRoundData(Denominations.ETH,Denominations.USD);returnprice;}/ * Returns the latest price /functiongetPrice(addressbase,addressquote)publicviewreturns(int){// prettier-ignore(/uint80 roundID/,intprice,/uint startedAt/,/uint timeStamp/,/uint80 answeredInRound*/)=registry.latestRoundData(base,quote);returnprice;}} Open in Remix What is Remix?

Solidity Hardhat Example

Note

You can find a working Feed Registry Hardhat projecthere . Clone the repo and follow the setup instructions to run the example locally.

Javascript

/* * THIS IS AN EXAMPLE CONTRACT THAT USES HARDCODED VALUES FOR CLARITY. * THIS IS AN EXAMPLE CONTRACT THAT USES UN-AUDITED CODE. * DO NOT USE THIS CODE IN PRODUCTION. /const web3=newWeb3("https://rpc.ankr.com/eth")const feedRegistryInterfaceABI=[{anonymous:false,inputs: [{indexed:true,internalType:"address",name:"accessController",type:"address"},{indexed:true,internalType:"address",name:"sender",type:"address"},],name:"AccessControllerSet",type:"event",}, {anonymous:false,inputs:[{indexed:true,internalType:"address",name:"asset",type:"address"},{indexed:true,internalType:"address",name:"denomination",type:"address"}, {indexed:true,internalType:"address",name:"latestAggregator",type:"address"},{indexed:false,internalType:"address",name:"previousAggregator",type:"address"}, {indexed:false,internalType:"uint16",name:"nextPhaseId",type:"uint16"},{indexed:false,internalType:"address",name:"sender",type:"address"},],name:"FeedConfirmed",type:"event",}, {anonymous:false,inputs:[{indexed:true,internalType:"address",name:"asset",type:"address"},{indexed:true,internalType:"address",name:"denomination",type:"address"}, {indexed:true,internalType:"address",name:"proposedAggregator",type:"address"},{indexed:false,internalType:"address",name:"currentAggregator",type:"address"}, {indexed:false,internalType:"address",name:"sender",type:"address"},],name:"FeedProposed",type:"event",},{anonymous:false,inputs:[{indexed:true,internalType:"address",name:"from",type:"address"}, {indexed:true,internalType:"address",name:"to",type:"address"},],name:"OwnershipTransferRequested",type:"event",},{anonymous:false,inputs: [{indexed:true,internalType:"address",name:"from",type:"address"},{indexed:true,internalType:"address",name:"to",type:"address"},],name:"OwnershipTransferred",type:"event",},{inputs: [],name:"acceptOwnership",outputs:[],stateMutability:"nonpayable",type:"function"},{inputs:[{internalType:"address",name:"base",type:"address"},{internalType:"address",name:"quote",type:"address"}, {internalType:"address",name:"aggregator",type:"address"},],name:"confirmFeed",outputs:[],stateMutability:"nonpayable",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},],name:"decimals",outputs:[{internalType:"uint8",name:"",type:"uint8"}],stateMutability:"view",type:"function",},{inputs: [{internalType:"address",name:"base",type:"address"},{internalType:"address",name:"quote",type:"address"},],name:"description",outputs: [{internalType:"string",name:"",type:"string"}],stateMutability:"view",type:"function",},{inputs:[],name:"getAccessController",outputs:[{internalType:"contract AccessControllerInterface",name:"",type:"address"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},{internalType:"uint256",name:"roundId",type:"uint256"},],name:"getAnswer",outputs: [{internalType:"int256",name:"answer",type:"int256"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},],name:"getCurrentPhaseId",outputs:[{internalType:"uint16",name:"currentPhaseId",type:"uint16"}],stateMutability:"view",type:"function",},{inputs: [{internalType:"address",name:"base",type:"address"},{internalType:"address",name:"quote",type:"address"},],name:"getFeed",outputs:[{internalType:"contract AggregatorV2V3Interface",name:"aggregator",type:"address"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},{internalType:"uint80",name:"roundId",type:"uint80"},],name:"getNextRoundId",outputs: [{internalType:"uint80",name:"nextRoundId",type:"uint80"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},{internalType:"uint16",name:"phaseId",type:"uint16"},],name:"getPhase",outputs:[{components: [{internalType:"uint16",name:"phaseId",type:"uint16"},{internalType:"uint80",name:"startingAggregatorRoundId",type:"uint80"}, {internalType:"uint80",name:"endingAggregatorRoundId",type:"uint80"},],internalType:"struct FeedRegistryInterface.Phase",name:"phase",type:"tuple",},],stateMutability:"view",type:"function",},{inputs: [{internalType:"address",name:"base",type:"address"},{internalType:"address",name:"quote",type:"address"},{internalType:"uint16",name:"phaseId",type:"uint16"},],name:"getPhaseFeed",outputs: [{internalType:"contract AggregatorV2V3Interface",name:"aggregator",type:"address"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},{internalType:"uint16",name:"phaseId",type:"uint16"},],name:"getPhaseRange",outputs: [{internalType:"uint80",name:"startingRoundId",type:"uint80"},{internalType:"uint80",name:"endingRoundId",type:"uint80"},],stateMutability:"view",type:"function",},{inputs: [{internalType:"address",name:"base",type:"address"},{internalType:"address",name:"quote",type:"address"},{internalType:"uint80",name:"roundId",type:"uint80"},],name:"getPreviousRoundId",outputs: [{internalType:"uint80",name:"previousRoundId",type:"uint80"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},],name:"getProposedFeed",outputs:[{internalType:"contract AggregatorV2V3Interface",name:"proposedAggregator",type:"address"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},{internalType:"uint80",name:"_roundId",type:"uint80"},],name:"getRoundData",outputs:[{internalType:"uint80",name:"roundId",type:"uint80"}, {internalType:"int256",name:"answer",type:"int256"},{internalType:"uint256",name:"startedAt",type:"uint256"},{internalType:"uint256",name:"updatedAt",type:"uint256"}, {internalType:"uint80",name:"answeredInRound",type:"uint80"},],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},{internalType:"uint80",name:"roundId",type:"uint80"},],name:"getRoundFeed",outputs:[{internalType:"contract AggregatorV2V3Interface",name:"aggregator",type:"address"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},{internalType:"uint256",name:"roundId",type:"uint256"},],name:"getTimestamp",outputs: [{internalType:"uint256",name:"timestamp",type:"uint256"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"aggregator",type:"address"}],name:"isFeedEnabled",outputs: [{internalType:"bool",name:"",type:"bool"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},],name:"latestAnswer",outputs:[{internalType:"int256",name:"answer",type:"int256"}],stateMutability:"view",type:"function",},{inputs: [{internalType:"address",name:"base",type:"address"},{internalType:"address",name:"quote",type:"address"},],name:"latestRound",outputs: [{internalType:"uint256",name:"roundId",type:"uint256"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"}, {internalType:"address",name:"quote",type:"address"},],name:"latestRoundData",outputs:[{internalType:"uint80",name:"roundId",type:"uint80"},{internalType:"int256",name:"answer",type:"int256"}, {internalType:"uint256",name:"startedAt",type:"uint256"},{internalType:"uint256",name:"updatedAt",type:"uint256"}, {internalType:"uint80",name:"answeredInRound",type:"uint80"},],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"},

{internalType:"address",name:"quote",type:"address"},],name:"latestTimestamp",outputs:[{internalType:"uint256",name:"timestamp",type:"uint256"}],stateMutability:"view",type:"function",},{inputs:
[],name:"owner",outputs:[{internalType:"address",name:"",type:"address"}],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"},
{internalType:"address",name:"quote",type:"address"},{internalType:"address",name:"aggregator",type:"address"},],name:"proposeFeed",outputs:[],stateMutability:"nonpayable",type:"function",},{inputs:
[{internalType:"address",name:"base",type:"address"},{internalType:"address",name:"quote",type:"address"},
{internalType:"uint80",name:"roundId",type:"uint80"},],name:"proposedGetRoundData",outputs:[{internalType:"uint80",name:"id",type:"uint80"},{internalType:"int256",name:"answer",type:"int256"},
{internalType:"uint256",name:"startedAt",type:"uint256"},{internalType:"uint256",name:"updatedAt",type:"uint256"},
{internalType:"uint80",name:"answeredInRound",type:"uint80"},],stateMutability:"view",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"},
{internalType:"address",name:"quote",type:"address"},],name:"proposedLatestRoundData",outputs:[{internalType:"uint80",name:"id",type:"uint80"},{internalType:"int256",name:"answer",type:"int256"},
{internalType:"uint256",name:"startedAt",type:"uint256"},{internalType:"uint256",name:"updatedAt",type:"uint256"},
{internalType:"uint80",name:"answeredInRound",type:"uint80"},],stateMutability:"view",type:"function",},{inputs:[{internalType:"contract
AccessControllerInterface",name:"_accessController",type:"address"}],name:"setAccessController",outputs:[],stateMutability:"nonpayable",type:"function",},{inputs:
[{internalType:"address",name:"to",type:"address"}],name:"transferOwnership",outputs:[],stateMutability:"nonpayable",type:"function",},{inputs:[],name:"typeAndVersion",outputs:
[{internalType:"string",name:"",type:"string"}],stateMutability:"pure",type:"function",},{inputs:[{internalType:"address",name:"base",type:"address"},
{internalType:"address",name:"quote",type:"address"},],name:"version",outputs:[{internalType:"uint256",name:"",type:"uint256"}],stateMutability:"view",type:"function",},]const
addr="0x47Fb2585D2C56Fe188D0E6ec628a38b74fCeeeDf"const feedRegistry=newweb3.eth.Contract(feedRegistryInterfaceABI,addr)const
LINK="0x514910771AF9Ca656af840dff83E8264EcF986CA"const
USD="0x0000000000000000000000000000000000000348"feedRegistry.methods.latestRoundData(LINK,USD).call().then((roundData)=>{// Do something with roundDataconsole.log("Latest Round
Data",roundData)})

Latest LINK/USD Price:Latest Price

Contract addresses

This section lists the blockchains that Chainlink Feed Registry are currently available on.

NetworkAddressEthereum Mainnet0x47Fb2585D2C56Fe188D0E6ec628a38b74fCeeeDf