

Chainlink

[Chainlink](#) is a widely-recognized Web3 services platform that specializes in decentralized oracle networks. It lets you build Ethereum and Arbitrum dApps that connect to a variety of off-chain data feeds and APIs, including those that provide asset prices, weather data, random number generation, and more.

Querying the price of ARB through Chainlink

Here's an example on how to use a price feed from Chainlink to query the current price of ARB on-chain. We'll use an interface provided by Chainlink that can be configured with the address of the proxy that holds the information we want to request, and wrap the operation in a contract.

Chainlink provides an npm package with the contracts needed to access their feeds. We first install that package in our project:

yarn add @chainlink/contracts To use a data feed, we retrieve the information through the AggregatorV3Interface and the proxy address of the feed we want to query.

import

"@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol" ; In this case, we want to obtain the current price of ARB in USD in Arbitrum One, so we need to know the address of the proxy that will provide that information. Chainlink maintains a list of price feed addresses [here](#) . For ARBUSDT, we'll use the address 0xb2A824043730FE05F3DA2efaFa1CBbe83fa548D6 .

We can now build the function to get the latest price of ARB. We'll use this example contract:

contract

ARBPriceConsumer

{ AggregatorV3Interface internal priceFeed ;

/ * Network: Arbitrum One * Aggregator: ARB/USD * Address: 0xb2A824043730FE05F3DA2efaFa1CBbe83fa548D6 */*
address

constant PROXY =

0xb2A824043730FE05F3DA2efaFa1CBbe83fa548D6 ;

constructor ()

{ priceFeed =

AggregatorV3Interface (PROXY) ; }

/ * Returns the latest price. */* function

getLatestPrice ()

public

view

returns

(int)

{ (/ uint80 roundID / , int price , / uint startedAt / , / uint timeStamp / , / uint80 answeredInRound /)

= priceFeed . latestRoundData () ; return price ; } } You can adapt this contract to your needs. Just remember to use the address of the asset you want to request the price for in the appropriate network, and to deploy your contract to the same network . Remember we have a [Quickstart](#) available that goes through the process of compiling and deploying a contract.

More examples

Refer to [Chainlink's documentation](#) for more examples of querying price feeds plus other data feeds available [Edit this page](#)
Last updated on Jan 27, 2025 [Previous API3](#) [Next Chronicle](#)