

# Trellor

[Trellor](#) is a decentralized oracle network that incentivizes an open, permissionless network of data reporting and validation, ensuring that any verifiable data can be brought on-chain. It supports basic spot prices, sophisticated pricing specs (TWAP/VWAP), Snapshot Vote Results, and custom data needs.

## Querying the price of ETH through Tellor

Here's an example of how to use a Tellor data feed to query the current price of ETH on-chain. The way it works is that a query is crafted asking for the price of one currency against another and sent to the oracle contract. If the information for that query is available, it will be returned. Oracle contracts can be found on the [Contracts Reference](#) page.

Tellor provides an npm package with the contracts needed to query the contract. We first install that package in our project:

```
npm
install usingtellor
```

Our function will just wrap the call to the Oracle contract with the query we are interested in. In this case, we want to obtain the "SpotPrice" of "eth" against "usd". We will request this information to the Arbitrum oracle contract `0xD9157453E2668B2fc45b7A803D3FEF3642430cC0`. We'll use this example contract:

```
contract
ARBPriceConsumer

is UsingTellor { /* * Network: Arbitrum One * Aggregator: ARB/USD * Address:
0xD9157453E2668B2fc45b7A803D3FEF3642430cC0 / constructor ( address
payable _tellerAddress )

UsingTellor ( _tellerAddress ) { }

/* * Returns the latest price.*/ function

getLatestPrice ( )

public

view

returns

( uint256 )

{ bytes
memory _queryData = abi . encode ( "SpotPrice" , abi . encode ( "eth" ,
"usd" ) ) ; bytes32 _queryId =
keccak256 ( _queryData ) ;

( bytes
memory _value ,
uint256 _timestampRetrieved )

= getDataBefore ( _queryId , block . timestamp -
20 minutes ) ; if
( _timestampRetrieved ==
0 )

return

0 ; require ( block . timestamp - _timestampRetrieved <
24 hours ) ; return abi . decode ( _value ,
```

( uint256 ) ) ; } } You can adapt this contract to your needs. Just remember to use the ticker of the assets you want to request the price for and to deploy your contract to the appropriate network, with the address of the Oracle contract in that network . Remember, we have a [Quickstart](#) available that goes through the process of compiling and deploying a contract.

## See also

- [Tellor's documentation](#)
- demonstrates how to query price feeds and other data feeds. [Edit this page](#) Last updated on Jan 27, 2025 [Previous](#) [How to use Supra VRF](#) [Next](#) [Contribute third-party docs](#)