Basic Request Model

Contracts overview

All source code is open source and available in theChainlink Github repository .

ChainlinkClient

ChainlinkClient is a parent contract that enables smart contracts to consume data from oracles. It's available in the Chainlink smart contract library which can beinstalled using the latest package managers .

The client constructs and makes a request to a known Chainlink oracle through thetransferAndCallfunction, implemented by the LINK token. This request contains encoded information that is required for the cycle to succeed. In theChainlinkClientcontract, this call is initiated with a call tosendChainlinkRequestTo.

To build your own client contract usingChainlinkClient, seeIntroduction to Using Any API , or view theChainlinkClient API Reference for theChainlinkClientcontract.

LINK Token

LINK is anERC-677 compliant token which implementstransferAndCall, a function that allows tokens to be transferred whilst also triggering logic in the receiving contract within a single transaction.

Learn more aboutERC-677 and the LINK token .

Operator Contract

Operatorcontracts are owned by oracle node operators, which run alongside offchain oracle nodes.

Request

The client contract that initiates this cycle must create a request with the following items:

- The oracle address.
- The job ID, so the oracle knows which tasks to perform.
- The callback function, which the oracle sends the response to.

To learn about how to find oracles to suit your needs, seeFind Existing Jobs .

Operator contracts are responsible for handling onchain requests made through the LINK token, by implementingonTokenTransferas aLinkTokenReceiver . Upon execution of this function, the operator contractemits anOracleRequesteventcontaining information about the request. This event is crucial, as it is monitored by the offchain oracle node which acts upon it.

Fulfillment

For fulfillment, the operator contract has afulfillOracleRequestfunction which is used by the node to fulfill a request once it has the result of the job. This function returns the result to theChainlinkClientusing the callback function defined in the original request.

Offchain oracle node

The offchain oracle node is responsible for listening for events emitted by its corresponding onchain smart contract. Once it detects anOracleRequestevent, it uses the data emitted to perform a job.

The most common job type for a Node is to make a GET request to an API, retrieve some data from it, parse the response, convert the result into blockchain compatible data, then submit it in a transaction back to the operator contract, using thefulfillOracleRequestfunction.

For more information on how to become a node operator, learn how torun a Chainlink node .

Consumer UML

Below is a UML diagram describing the contract structure ofATestnetConsumer, a deployed example contract implementingChainlinkClient.