

# #

Oracle

# #

## Introduction

The module combines `service` to achieve decentralized injection from trusted Oracles such as Chainlink to IRIShub Oracle. Each data collection task is called a feed, and its underlying implementation depends on the `service` module. Feed life cycle It is basically the same as the `Service RequestContext` (paused, running), and is used to store off-chain data on the chain through Oracle nodes. In addition, you can only operate `Feed` through your Profiler account, and you cannot delete it. You can only pause the feed. Including the following operations:

- Create Feed
- Start Feed
- Pause Feed
- Edit Feed

In addition to collecting data by creating a feed, the module also presets some aggregation functions, such as `avg`, `max`, `min`, etc., to process the collected data to meet various scenarios. The data collected by each `Feed` can only save the most recent 100 entries, and the rest will be deleted.

# #

## Process

The bottom layer of the module depends on the `service` module, so the premise of using this module is to execute the relevant functions of `Service`

- `CreateService`
- `definition`
- `BindService`

Specific instructions [service](#) . After completing the `service` related operations, start the `Oracle` process:

### 1. Create Feed

```
iris tx oracle create \ --feed-name= "test-feed" \ --latest-history= 10 \ --service-name= "test-service" \ --input = { request-data} \
--providers = "faa1hp29kuh22vpjlnctmym15s75evsnsd8r4x0mm,faa15rurzhkemsgfm42dnwhafjdv5s8e2pce0ku8ya" \ --
service-fee-cap= 1iris \ --timeout = 2 \ --frequency = 10 \ --threshold = 1 \ --aggregate-func= "avg" \ --value-json-path= "high" \
--chain-id= "irishub-test" \ --from = node0 \ --fees = 0 .3iris \ --broadcast-mode= block 2. Start Feed
```

After the `Feed` is created, the collection task is in the `paused` state, and no request is made to the service provider. You can start the scheduled task of the feed through `start` .

```
iris tx oracle start test-feed \ --chain-id= "irishub-test" \ --from = node0 \ --fees = 0 .3iris \ --broadcast-mode= block 3. Pause
Feed
```

Since `Feed` cannot be deleted once it is created, it will consume the balance of the owner's account until the balance is exhausted, and `Feed` will enter the `paused` state. To be able to pause the feed manually, you can use the `pause` command

```
iris tx oracle pause test-feed \ --chain-id= "irishub-test" \ --from = node0 \ --fees = 0 .3iris \ --broadcast-mode= block 4. Edit
Feed
```

You can use the `edit` command to edit an existing feed to change the data collection behavior of the feed.

```
iris tx oracle edit test-feed \ --latest-history= 5 \ --providers =
"faa1r3tyupskwlh07dmhjw70frxzaaaufta37y25yr,faa1ydahnhrhkjh9j9u0jn8p3s272l0ecqj40vra8h" \ --service-fee-cap= 1iris \ --
timeout = 6 \ --threshold = 5 \ --chain-id= "irishub-test" \ --from = node0 \ --fees = 0 .3iris \ --broadcast-mode= block Note that
if the latest-history at the time of creation is greater than the currently modified value, the oracle module will delete the extra
data.
```