

WHAT IS AN EYONCHS FEED IN REDIS ?

A HASH OF METADATA S

Key	Fields	Values
feed:(id)	id	1601125200
	engine	g
	name	
	datatype	0
	userid	1
	public	
	tag	

MANDATORY

AN ENTRY IN THE

user:(userid): Feeds Key

which is a
sorted set
gathering all
the feed ids
belonging to
the user

start_time & interval are mandatory for PHP FINA but for
a feed already sorted as a timeserie, they are not

USER 1 LOGGED IN \Rightarrow feed / list.json

The screenshot shows a web browser window with the address bar displaying `localhost/emoncms/feed/list.json`. The page content shows a JSON feed with the following data:

```
{
  "interval": 10,
  "2": {
    "id": "1601125200",
    "tag": "weather_forecasts",
    "engine": "g",
    "name": "OWMHourly_2020_09_26_15_00_00_+0200",
    "datatype": "0",
    "userid": "1",
    "public": "",
    "time": null,
    "value": null,
    "start_time": 0,
    "interval": 1
  }
}
```

A red arrow points from the `"id": "1601125200"` field in the JSON data to the text "The Redis feed created".

\Rightarrow feed / data.json ? id = 1601125200



HOW DOES THE ENGINE RETRIEVE THE DATAS ?

PHPPFINA

$$\text{time} = \text{start} + i * \text{interval}$$

$$\text{pos} = \frac{\text{time} - \text{meta}[\text{start_time}]}{\text{meta}[\text{interval}]}$$

\Rightarrow data[time] can be found in the dat file
at position pos

\Rightarrow UNPACK

PHPTIME SERIES

start = 0

end = File size / 9 (each value is 9 bytes)

For i = 0 to 30

mid = start + $\frac{\text{end} - \text{start}}{18} * 9$

if time == time[mid] return mid

if end - start == 9 return mid - 9

if time > time[mid] start = mid else end = mid

⇒ data[time] can be found in the dat file
at position mid

⇒ UNPACK

REDIS BUFFER

ONLY USED FOR DISK-WRITING PURPOSES

WHY NOT USING IT TO RENDER SMALL SIZE
SAMPLES, SUCH AS METEO FORECASTS ?

Numpy Array

time	Value

⇒ Flattened
+
converted
to binary string
(python tobytes() method
from the numpy package)

⇒ feed : (id):buffer

⇒ A PYTHON SERVICE FOR
SERVICE RUNNER ?

READING THE REDIS BUFFER WHEN
IT IS SUCH A BINARY KEY

A SIMPLE PROCESS

1) A REDIS GET \rightarrow \$result

2) UNPACK \rightarrow \$tab = unpack('d*', \$result, 8)

\Rightarrow \$tab[0] = time stamp
\$tab[1] = value

DONE !