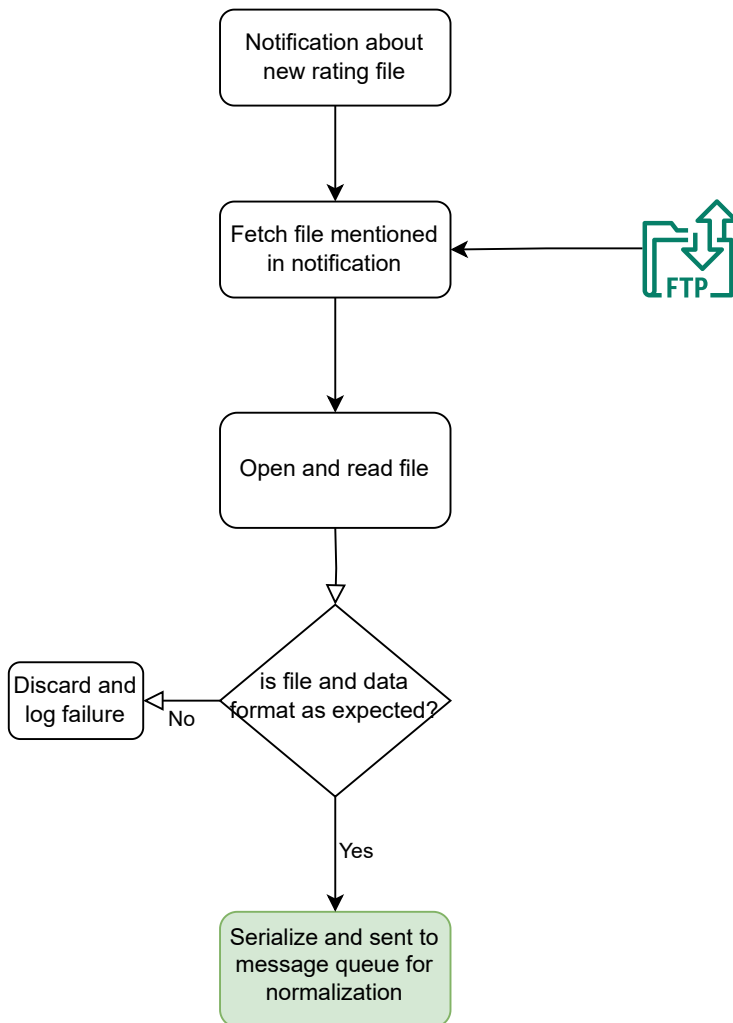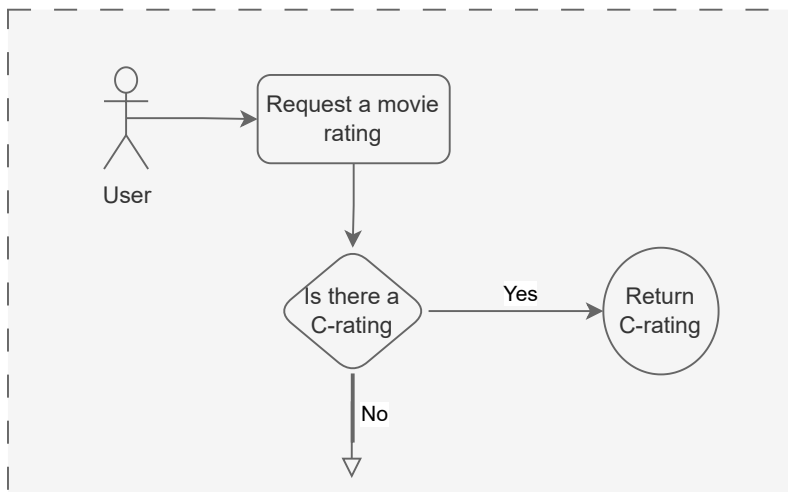# Periodic Data Fetching from Offline Feed

A cron job is triggering the fetching from the FTP server. This could also be the source of the first data population of our system.
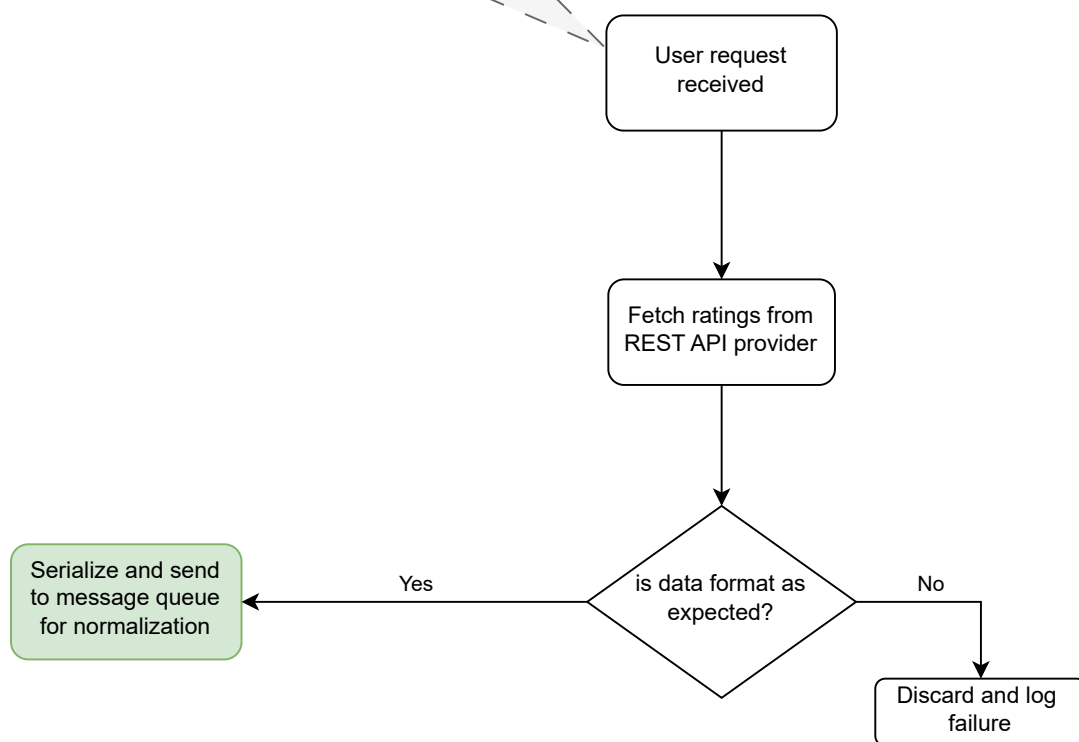
Fetch new or updated file from FTP server

Is there a new file?

No

Yes

Open and read file

is file and data format as expected?

Discard File

No

Yes

Serialize data and send for normalization

```
  ┌─────────────────────┐
  │  Notification about  │
  │   new rating file    │
  └─────────────────────┘
             │
             ▼
  ┌─────────────────────┐
  │ Fetch file mentioned │ ◄───────────  [FTP]
  │   in notification    │
  └─────────────────────┘
             │
             ▼
  ┌─────────────────────┐
  │   Open and read file │
  └─────────────────────┘
             │
             ▼
            ╱ ╲
           ╱   ╲
  ┌──────┐╱ is file and╲
  │Discard│◄  data format │  No
  │and log │╲ as expected?╱
  │failure │ ╲          ╱
  └──────┘   ╲        ╱
             ╲      ╱
              Yes
               │
               ▼
  ┌─────────────────────┐
  │ Serialize and sent to│
  │ message queue for    │
  │ normalization        │
  └─────────────────────┘
```
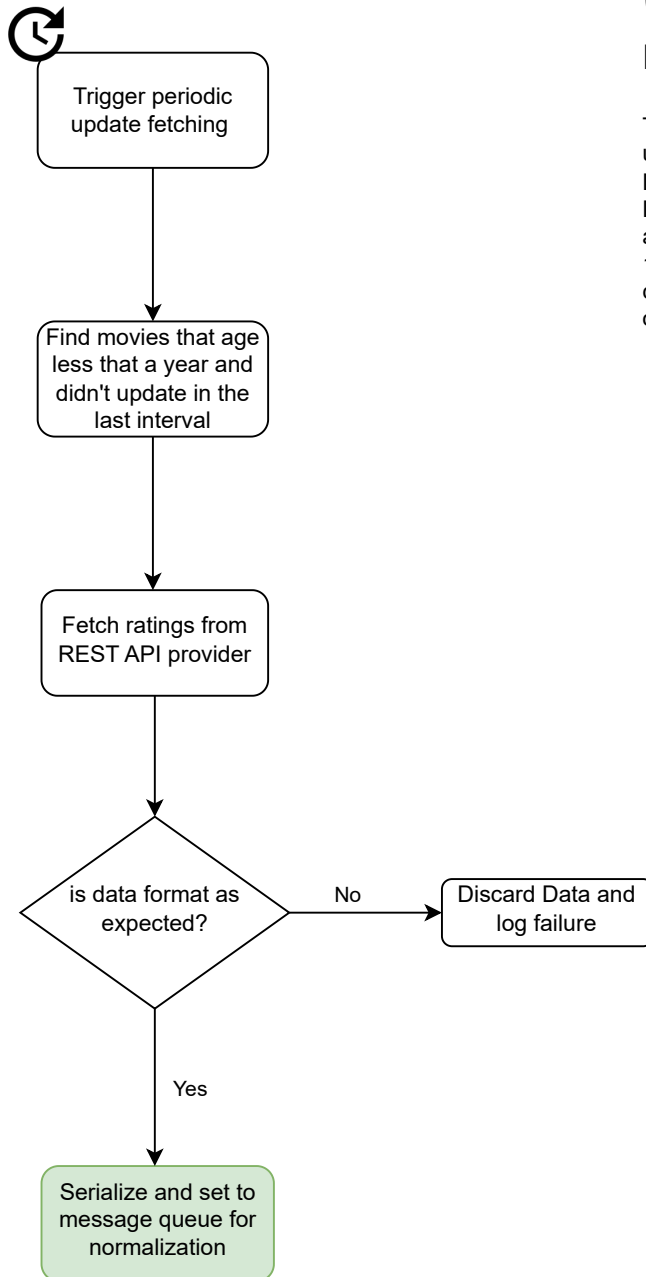
Online feed

The fetching here is based on an RSS feed (or equivalent) and is based on an assumption that our system will be notified upon updates.

Fetch upon user request

In order to generate more value, user requests for movies that haven't been processed yet will be taken into concentration. The request will be processed asynchronously and the user will be informed that their request is pending.

User

Request a movie rating

Is there a C-rating

Yes → Return C-rating

No

User request received

Fetch ratings from REST API provider

is data format as expected?

Yes → Serialize and send to message queue for normalization

No → Discard and log failure

Updates of new movies.

This is a periodic fetching of updates of existing movies in the DB. It requires interaction with the DB and is based on the assumption that new movies (< 1year old) will have their ratings change radically within the course of on year.

Trigger periodic update fetching

Find movies that age less that a year and didn't update in the last interval

Fetch ratings from REST API provider

is data format as expected?

No → Discard Data and log failure

Yes

Serialize and set to message queue for normalization

Populate Normalized DB

To populate the DB with the data from the providers, we must first normalize the data, as well as deduplicate and provide a common movie id. When a new entry is confirmed, we can also check for missing providers of the same movie and trigger API requests.

is message queue empty

Yes

No

Get next rating from message queue

Get Unique ID for movie from 3rd party provider

Normalize data(0-100)

Write or update (if already exists) rating

Yes

writing was successful?

No

Log failure