

Email retry mechanism

A different topic with a new retrieval consumer to add an entry in the email reminder read model

Snapshot, backup & Clean-up strategies for Reminders write model

Prevent race condition in email read model when update message and deleting entry

If the write model isn't enough, an audit write model, for every job, consumer and processor to make clear what has been executed (Besides log traces and metrics)

Model HTTP 4xx, 5xx and other error codes

Optimise daily jobs partitioning or sharding the data layer, when needed

Reminder processor retry mechanism

Back-pressure mechanism for Dashboard APP

For geographical optimization, this could be deployed in different regions. For availability, in different zones.

```
paths:
  /me/reminder:
    post:
      summary: Creates a reminder.
      requestBody:
        required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              utcDateTime:
                description: the current timestamp represented in UTC
                type: string
                format: date
                example: '2017-07-21T17:32:28Z'
              localTimeOffset:
                description: the local timezone offset from UTC
                type: string
                example: 'UTC+2'
            note:
              description: the reminder content / text
              type: string
              example: 'Remember to request a copy of the invoice'
```

```
RecurrentReminder:
  type: object
  properties:
    frequency:
      description: the time frequency unit for the recurrent reminder
      type: string
      enum: [DAILY, WEEKLY, MONTHLY or YEARLY]
    interval:
      description: how many days, weeks, month or years should pass between each notification
      type: integer
      format: int64
      minimum: 1
    required:
      - utcDateTime
      - localTimeOffset
```

```
responses:
  '201':
    description: The reminder id
    content:
      application/json:
        schema:
          type: object
          properties:
            id:
              type: string
              format: uuid
              example: '123e4567-e89b-12d3-a456-426614174000'
```

```
sendByemail:
  description: a flag to send this reminder the first time by email
  type: boolean
  example: true
recurrentReminderConfig:
  $ref: '#/components/schemas/RecurrentReminder'
required:
  - utcDateTime
  - localTimeOffset
  - note
```

```
paths:
  /me/reminder:
    post:
      summary: Creates a reminder.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                utcDateTime:
                  description: the current timestamp represented in UTC
                  type: string
                  format: date
                  example: '2017-07-21T17:32:28Z'
                localTimeOffset:
                  description: the local timezone offset from UTC
                  type: string
                  example: 'UTC+2'
                note:
                  description: the reminder content / text
                  type: string
                  example: 'Remember to request a copy of the invoice'
                sendByemail:
                  description: a flag to send this reminder the first time by email
                  type: boolean
                  example: true
                recurrentReminderConfig:
                  $ref: '#/components/schemas/RecurrentReminder'
      required:
        - utcDateTime
        - localTimeOffset
        - note
```

```
responses:
  '201':
    description: The reminder id
    content:
      application/json:
        schema:
          type: object
          properties:
            id:
              type: string
              format: uuid
              example: '123e4567-e89b-12d3-a456-426614174000'
RecurrentReminder:
  type: object
  properties:
    frequency:
      description: the time frequency unit for the recurrent reminder
      type: string
      enum: [DAILY, WEEKLY, MONTHLY or YEARLY]
    interval:
      description: how many days, weeks, month or years should pass between each notification
      type: integer
      format: int64
      minimum: 1
    required:
      - utcDateTime
      - localTimeOffset
```



API

POST /me/reminder
 PATCH /me/reminder/:uuid
 PATCH /me/reminder/:uuid/done
 DELETE /me/reminder/:uuid

Assuming an auth header with user UUID claim in it

The service is deployed in EKS (AWS K8s), so elastic scalability is in place

Assuming an auth header with user UUID claim for /me endpoints

For email and daily reminder jobs, k8s jobs will be used, with 1 short-lived pod at once to avoid duplicate reminders

Using HTTP API instead of SMTP for sending emails as it's good enough for our use case: It's straightforward and there isn't a need for fine-grained control or additional redundancy assurances

Only one reminder per rule will be published according to frequency as the dashboard will keep the living (not done) reminders or (preferred) use the "Occurrences" field

I don't love the sent counter, but It's the first thing that came to my mind for living (not done) reminders

For sure with users hitting this we will get a better solution

Indexes as of
now:
ReminderId,
UserId,
UtcDateTime

Using
PostgreSQL
/ AuroraDB



Reminders write model

ReminderId	UserId (Aggregate)	EventType	Payload	InsertTimestamp
UUID	UUID	enum (CREATE, UPDATE, DELETE, DONE)	json	timestamp

json type
payload instead
of jsonb as this
isn't for read
purposes

jsonb payload can be
used when migrating
to BI / Analytics / DWH
side of things



Email read model

ReminderId	UserId (Aggregate)	LocalTimeOffset	UtcDateTime	Note	Recurrent	Frequency	Interval	InsertTimestamp	UpdateTimestamp
UUID	UUID	varchar(5)	timestamp	text	boolean	enum (DAILY, WEEKLY, MONTHLY, YEARLY)	integer	timestamp	timestamp

Only used for
conversion, if
needed



Recurrent rule reminder
read model

ReminderId	UserId (Aggregate)	UtcDateTime	Note	Recurrent	Ocurrences	Frequency	Interval	UpdateTimestamp
UUID	UUID	timestamp	text	boolean	integer	enum (DAILY, WEEKLY, MONTHLY, YEARLY)	integer	timestamp

Request Topic



Create
Message

Based on
same content
as the request
payload + type

Update
Message

Based on
same content
as the request
payload + type

Delete
Message

```
{
  "type": "DELETE",
  "reminderId": "<UUID>",
  "userId": "<UUID>"
}
```

Done
Message

```
{
  "type": "DONE",
  "reminderId": "<UUID>",
  "userId": "<UUID>"
}
```

Email topic



Email
Reminder
Message

```
{
  "reminderId": "<UUID>",
  "userId": "<UUID>",
  "note": "Message body..."
}
```

Reminder topic



Note
Reminder
Message

```
{
  "reminderId": "<UUID>",
  "userId": "<UUID>",
  "note": "Message body...",
  "Occurrences": 3
}
```



