

Petit chat

Requirements

- send notifications to users
- chat with 2 or more users
- use an sso to connect
- messages can have
 - text OK
 - attachments OK
 - reactions OK
- scylladb as db

Notifications

Use external notification gateway -> talk to it using a queue system // topic by userId -> can have many workers

Push notifications SSE when app is opened -> requires internal notification system to handle distributed app -> Queue / reactive messaging

SSO

-> use keycloak

Messages / Signal (to support more use cases)

-> use cassandra udt / collections to store the infos ? -> handle the query get all files in chat -> use S3 to store attachments -> [seaweedfs](#) -> dedupe using hashing ?

Queries

db: scylladb

- find user by username -> ok but not paged
- find all my chats paged -> ok but not paged
- get messages from chat at offset -> ok
- find in chat by message content -> need text indexer
 - should implem meili search first for basic use -> OK
 - implem elastic or more powerful with sharding later

For each message -> handle reactions -> ok not paged for now -> handle views (wich users saw the message) / can handle the deliver state as well -> ok, not paged for now

Optimizations ?

store the count of reactions on the signal directly -> use atomic increment / decrement

Signal types

- msg
- joined / leaved
- calls

TODO

- add file attachment support -> WIP can upload the files -> need to assert how to view the files
- add signal types
 - joined / leaved -> WIP, need test
 - calls
- add support for current status
- add support for share point

Architecture

```
client <=> network-provider.gateway
network-provider.gateway <=> api.controller
network-provider.gateway -> seaweed.volume
network-provider.gateway -> sso.controller

api -> sso.controller
api.service <=> queue
api.service -> seaweed.master
api.service -> seaweed.volume
api.service -> text-indexer
api.repository -> db-cluster.db

network-provider {
  gateway
}

seaweed {
  volume
  master
  master -> volume
}

api {
  controller
  service
  repository
  controller -> service
```

```
    service -> repository
  }

  sso {
    controller
    postgres
    controller -> postgres
  }

  db-cluster {
    db
    manager
    manager -> db
  }

# d2-vscode can syntax highlight nested markdown correctly.
y: |`md

`|
```