**Tài liệu dự án tuyển cộng tác viên**

**Câu hỏi 1:** Mày muốn gì? Dự án này mày muốn làm nó với mục đích gì?

* TL: Tao thật sự không biết nữa! Chán quá! Không code thật chán! Tao chỉ muốn có một dự án để biết thứ giỏi nhất của mình không phế!

**Câu hỏi 2:** Vậy thì code đi. Nói tao nghe: dự án này mày muốn tập trung vào gì?

* TL: Tao muốn tập trung vào blog, vào chia sẻ, vào tìm kiếm.

Có nghĩ là mày muốn dự án tuyển cộng tác viên như 1 nơi chia sẻ niềm vui từ thiện, các dự án tốt, hay, được hỗ trợ bằng một hệ thống tìm kiếm theo **tag** chuẩn xác và nhanh chóng.

Mày có biết **nội dung** của một Blog: không chỉ có **chuỗi String** mà còn có **hình ảnh**, biểu tượng cảm xúc (sĩ nhiên là cần cái này, nếu không blog chỉ có chữ và chữ khá khô khan), bảng. Và còn có thể có biểu thức toán học và hình học.

* Dự án web tuyển cộng tác viên không tập trung vào sơ đồ dữ liệu, sơ đồ mối quan hệ trong database mà tập trung vào cấu trúc dữ liệu (cụ thể là cấu trúc nội dung của blog trong bảng Blog).
* Ta khuyên mày nên sử dụng elastichsearch để xây dựng công cụ tìm kiếm.

(Ý tưởng ngày: 09/11/2023 tạm thời là vậy đã)

11/11/2023 – làm chức năng của người dùng thông thường được trình bày trong file ‘database\_v1’.

# Chức năng thông báo:

## Giai đoạn 1 (24/11/2023): chức năng thông báo khi có bình luận trong bài viết

**Ý tưởng:** Mọi thông tin đều phải lưu trữ trong database kể cả số lượng thông báo chưa đọc. Redis sẽ lưu cache thông báo chưa đọc để tăng tốc độ truy vấn:  
Vậy ta sẽ có sơ đồ như sau:

Database

BE

Redis

request

Lấy số lượng thông báo

Gửi trả số lượng thông báo chưa đọc

Lưu xuống redis

**Câu hỏi 1:** Vậy Service nào của BE sẽ đóng vai trò cập nhật số lượng thông báo chưa đọc?

**Câu hỏi 2:** Cấu trúc bảng Comments hiện tại như sau:

Id | user\_id | blog\_id | commentLevel| comment| comment\_parent\_id

Làm sao mà ta biết được comment nào chưa đọc? và comment như nào thì thông báo? Comment trả lời 1 user cụ thể và comment mới xuất hiện trong bài viết cho tác giả à?

**Trả lời 1:** Để trả lời câu hỏi này thì rất khó. Ta có 2 kiểu cập nhật số lượng thông báo chưa đọc:

1. Khi sự kiện tạo thông báo xảy ra: ta + 1 trực tiếp vào số lượng thông báo chưa đọc.
2. Khi sự kiện tạo thông báo xảy ra: chèn thông báo vào bảng Notifications. Sau đó, ta lấy ra số lượng thông báo chưa đọc ở bảng Notifications và cập nhật lại số lượng thông báo chưa đọc.

Cách 1: cho phép giảm tải áp lực lên hệ thống vì không phải lấy nhiều dữ liệu, nhưng độ chính xác không cao và mức độ tập trung code kém do sự kiện dẫn đến tạo thông báo có thể phát sinh ở nhiều service.

Cách 2: thay vì để cho nhiều service cùng làm việc cập nhật thông báo chưa đọc, ta tập trung code ở 1 service **lắng nghe** các thay đổi trên các bảng trong database.

**Trả lời 2:** bảng Comment thì có liên quan gì. Ta cần tạo một bảng mới là Notification để ghi chép các thông báo và bảng Comment cũng **không cần** bổ sung thêm trường ‘**đọc nội dung hay chưa?**’vì chỉ cần người dùng bấm xem danh sách thông báo là **auto đánh dấu** phần thông báo câu trả lời đã được đọc như **youtube**.

Vậy thì làm sao thằng service lắng nghe các thay đổi trên các bảng trong database biết được người dùng đã lấy danh sách thông báo. Không lẽ lại làm thêm cho chức năng này phần cập nhật thông báo 🡪 có 2 nơi cập nhật thông báo đã đọc. Thế không ổn đâu.

Đọc binlog thì bó tay nhận biết được phần này, mà không đọc binlog thì phải chia phần cập nhật thông báo trên nhiều service 🡪 khó quản lý lắm!

Mà nghĩ lại thì binlog chỉ có thể cập nhật được thông báo chưa đọc nếu các bảng không có trường gì để nhận biết thông báo liên quan đã được xem.

Vậy là nhiều servie cập nhật **xem rồi**, binlog cập nhật **chưa xem** à? Đéo đâu, bảng Notifications có hẳn một trường để nhận biết thông báo đã đọc chưa mà! Còn cập nhật số lượng thông báo chưa đọc là việc của binlog. 🡺 nhiều service cập nhật trường “**xem rồi**” của Notification, binlog thì cập nhật **số lượng**.

Ngày 17/12/2023 – làm chức năng của công tác viên trong “database\_v1”

**Câu hỏi 1:** Ta có blogKey đóng vai trò không khác gì khóa chính – id, nhưng ngắn gọn hơn nhằm mục đích tạo url truy cập dễ dàng hơn mà không cần phải viết id dài dòng và khó nhớ. Hỏi blogKey nên do người dùng tạo hay tự tạo bằng máy?

**Trả lời 1:** Lần lượt xét từng trường hợp:

* Người dùng tự tạo. Ví dụ độ dài blogKey > 5 và nhỏ hơn 8, thì ban đầu khi số lượng bài viết còn ít thì người dùng còn dễ tạo khóa, đến khi số lượng bài viết lớn thì có khi không thể tạo nổi khóa ấy vì **mức độ trùng là rất cao**.
* Tự tạo bằng máy thì phải dựa vào cái gì, làm sao cho key dễ nhớ để dễ tìm kiếm?
* Dựa vào loginName ??? – không thể trùng mã id nhưng ngáo mới làm vậy? 🡪 như thế khác đéo nào vạch áo cho người xem lưng.
* Dựa vào gmail ??? - theo tham khảo trên facebook thì chỉ bạn bè với nhau là có thể nhìn thấy gmail của nhau trên đường dẫn khi truy cập vào trang cá nhân.
* Dựa vào tag và tiêu đề của bài viết – trùng thông tin thì tăng số thứ tự. *Ban đầu tag của bài viết vẫn chưa phát triển nên tạm thời bỏ qua*, dùng tiêu đề thì phải tóm gọn lại thế nào để ra blogKey. Nhớ lại bảng blog của mày có các trường gì?

Id | createTime | updateTime | blogKey | title | majorImgUrl | thumbnail | content | views | rate

**Câu hỏi 2:** phân loại bài viết đâu? Web chia sẻ các hoạt động tình nguyện mà cũng cần phân loại à? Phân loại thì phân loại thế nào?

**Trả lời câu hỏi 2**: Vậy hoạt động tình nguyện có bao nhiêu loại? Bài viết chia sẻ hoạt động tham gia tặng quà hay dạy học hay tuyển tình nguyên viên (cái này có chứ, đây 1 trong các mục đích của ta mà) 🡺 có cần phân loại bài viết. Tạm thời ta có các phân loại sau: (*theo hoạt động tình nguyện*): hỗ trợ người khiếm thị | hỗ trợ trẻ em vùng cao | chất độc màu da cam | … (*theo mục đích bài viết*): lan tỏa yêu thương | dạy học | tuyển tình nguyện viên | …

**Tiếp tục trả lời câu hỏi 1:** blog bây giờ sẽ bổ sung thêm tag (phân loại) và blogKey sẽ được tạo theo tag:

Blog

Tag

Phân loại

1,n

0,n

Các tag hiện tại bao gồm:

* Tag hoạt động tình nguyện:

1. Giúp đỡ học sinh bị khiếm thị - HSKT
2. Giúp đỡ học sinh vùng cao – HSVC
3. Giúp đỡ các trường hợp có hoàn cảnh khó khăn – HCKK
4. Giúp đỡ nạn nhân bị chất độc màu gia cam – CDMDC

* Tag mục đích đăng bài:
  1. Lan tỏa yêu thương – LTYT
  2. Dạy học – STU
  3. Tuyển tình nguyện viên - TTNV

Vậy tag có phân theo thứ tự chính phụ không? Theo tao thì hiện giờ thì chưa cần, nhưng sau này sẽ bổ sung sau.

🡺blogKey sẽ được tạo như sau: ngẫu nhiên tag theo hoạt động tình nguyện + ngẫu nhiên tag theo mục đích đăng bài + STT.