**project tuyển công tác viên.**

Form khóa key của redis:

<môi\_trường>:<module\_name>:<class\_name>:<tên\_khóa\_1>:<tên\_khóa\_2>.

Form chung của các value để phát hiện sự thay đổi của hệ thống trong redis là:

[msg@tenantId](mailto:msg@tenantId)=${tenantId}.

msg = gson.toJson(new OpLogRecord());

@OpLog:

* detail - tiêu đề sự thay đổi.
* describe - mô tả chi tiết sự thay đổi.
* bizId - khóa chính.
* bizTable - tên bảng.
* opType - loại thay đổi: insert, update, delete.

OpLogInterceptor chỉ gửi về gson.toJson(new OpLogRecord()).

1. Vậy tenantId ta lấy từ đâu?

**TL**: từ ThreadLocal, vậy cấu trúc nó ra làm sao.

Lớp nào sẽ chịu trách nhiệm lưu tenanId vào ThreadLocal.

Cá nhân tao cho rằng ở CustomRequestInterceptor.

nó lưu vào luồng trong module controller, nhưng ta xử lý @OpLog ở module service/module controller tùy thuộc vào queueName. Vậy làm sao ta biết tenantId chuẩn?

Chỉ còn cách lấy từ gson.toJson(new OpLogRecord()) - không, sai rồi!!!

Vậy ta lấy thông tin tenantId ở phía module service bằng cách nào?

Hãy nhớ thế này, ta gửi và nhận kết quả @OpLog ở module Controller → lưu vào ThreadLocal ở CustomRequestInterceptor.

2. Nếu ta lưu tenantId vào 1 đối tượng và chuyển đối tượng đó sang Json thì lúc chuyển từ Json sang đối tượng, thì ta biết ép kiểu đối tượng gì để lấy ra tenantId.

**TL**: Vậy thì ta không lưu tenantId vào Object nữa → lưu vào ThreadLocal tượng ứng thôi.

UserId, ta lấy từ đâu?

Trong dự án Fschool, ta lấy từ ThreadLocal của OpLogInterceptor, nếu như thế ta sẽ gặp vấn đề phải gán userId vào ThreadLocal tương ứng, rất phiền phức.

Tao quyết định, tao sẽ lưu vào session cơ.

BinlogeventListener làm gì?

1. kết nối binlog mysql - hiển nhiên.

2. cài đặt client.registerEventListener()

Chỉ những table được liệt kê trong DataSourceConf mới có các thao tác xử lý khi có 1 sự kiện xuất hiện.

Event → Xử lý → lưu vào database.

Hiện tai chức năng binlog của ta sẽ cố định bảng nào cần ghi lại thông tin thao tác, khi khởi động chương trình. Vậy liệu ta có cần thêm chức năng, trong thời gian run-time, admin được thêm động hoặc xóa các bảng cần ghi lại hay không?

TL: không cần. vì danh sách các bảng, ta lấy từ các file config mà!

Làm sao ta xác định được bảng nào là cần ghi lại, bảng nào là không ghi lại!

1. Liệu có cần phân chia bảng cần ghi, bảng không cần ghi không?

→ vì dữ liệu ở trong database cái nào cũng quan trọng mà

2. Thật sự không có giải pháp, vì 2 sự kiện gửi đến, ta đều không thể xác định được mới quan hệ giữa chúng.

Vl, lấy ra thông tin thao tác → xử lý → viết lại vào database => tạo thành vòng lặp không thể dừng => lấy thông tin binlog, thì không cần phải ghi lại vì:

1. lúc thao tác thêm, xóa, sửa vào → ta hoàn toàn có thể “giao dịch” được luôn vào bảng mình cần ghi thao tác. (ngày 09/11/2023 – nhận xét: lý do ta cần riêng 1 module để đọc binlog vì module này sẽ được triển khai trên server riêng không liên quan đến các module khác. Thứ 2: module đọc binlog này sẽ đọc trên 1 database riêng).

2. Gây lên áp lực lên database không đáng có.

Ta hoàn toàn có thể, tạo mô hình database master-slave, đọc ở slave, ghi ở master → không gây ra vòng lặp.

Nhưng làm binlog như vậy, thì có lợi gì!

**Không, ta phải áp dụng binlog vào vấn đề tích hợp → tạm thời xây dựng bộ khung → sau giải quyết tiếp.**

**Biểu thức tạo khóa chính** (8 bytes = 64 bits) như sau:

1. Độ dài của khóa chính phụ thuộc vào độ dài timestamp: timestamp = System.currentTimeMillis().
2. Id của hệ thống – systemId (có độ dài 4 bytes), được tạo bằng cách:
   1. lấy ra địa chỉ ip của hệ thống (xxx.xxx.xxx.xxx)
   2. Tách chuỗi ip dựa theo dấu “.” 🡪 được 1 mảng 4 chữa số.
   3. Id = 427546\*arr[0] + 120088\*arr[1] + 20275\*arr[2] + 887246\*arr[3].
3. Bộ đếm giá trị - incrementer (2 bytes).

**Công thức**: timestamp | systemId | incrementer