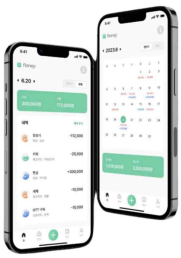
함께하는 금융 서비스, 공유 가계부 Floney

프로젝트 소개

사람들과 함께 사용한 돈의 흐름을 알기 쉽게 돕는 공유 가계부 서비스

런칭 일자 : 2023-12월 / 현 유저 수 : 약 600명



사용 기술

- Java
- Spring Boot, Spring MVC
- Spring Data JPA, QueryDSL
- MySQL
- AWS EC2, RDS

프로젝트 기록

'Floney' 카테고리의 글 목록 (tistory.com)

프로젝트 정보

기간 : 2023.04. ~ ing

개발팀 : IOS 1명 / BE 2명

프로젝트 주소

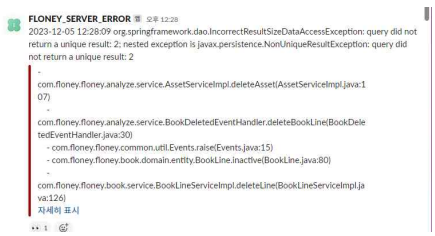
깃 : https://github.com/Floney-2023/Floney-Server

IOS 다운로드 링크

프로젝트 진행 내용

1. Upsert 문법을 통해 동시성 제어

문제 발생



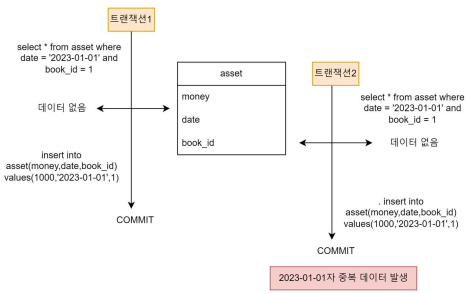
운영 서버와 연동된 슬랙 에러 알람 발생

자산 데이터는 가계부 별로 날짜가 유니크하게 들어가야 함.

그러나, **동일한 날짜의 자산 데이터가 2개씩 생성**

→ 에러 발생

원인

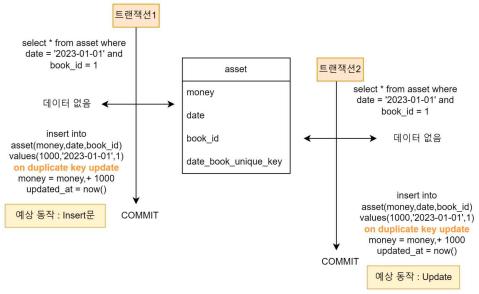


현재 DB의 트랜잭션 격리 수준은 **Repeatable Read**로 설정 됨

트랜잭션 1번이 book_id와 date를 기준으로 데이터 검색 후 없다면 데이터 삽입 후 커밋

만일 트랜잭션 1번이 커밋 되기 전, 2번 트랜잭션이 시작된 후 커밋 시, **중복 데이터가 발생**

해결 방안



PK 혹은 Unique Key 중복 시, insert가 아닌 update를 실행하는 mysql **Upsert 문법(on duplicate key update)**사용.

1. date와 book을 기준으로 복합 unique key로 설정
2. mysql에서 제공하는 upsert문법을 통해 트랜잭션2번이 Commit시 이미 트랜잭션 1번으로 인해 데이터가 존재하므로, update가 됨.

2. 상속을 통한 카테고리 기능 구현

요구 사항

1. 카테고리별 계층(부모 - 자식) 존재
2. 카테고리는 기본 제공 카테고리 와 가계부 별 사용자 정의 카테고리 두 가지 종류 존재

```java
@NoArgsConstructor(access = AccessLevel.PROTECTED)
@Getter
@Entity
public class Category extends BaseEntity {

    private String name;

    @ManyToOne
    private Book bookId; //가계부 id용

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "parent_id")
    @OnDelete(action = OnDeleteAction.CASCADE)
    private Category parent; //부모 카테고리

    @OneToMany(mappedBy = "parent")
    private List<Category> children = new ArrayList<>(); //자식 카테고리
}
```

[그림1] Self Referencing을 사용한 Category Entity

요구사항 1번인 계층형 카테고리 구현을 위해 **Self Referencing**을 사용하여 **Category Entity**구성

요구 사항 2번 구현 시, **문제 발생**

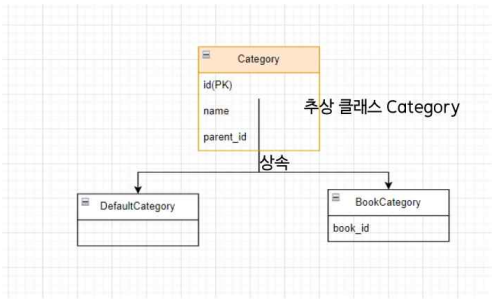
1. 탐색 비용 증가 **:** 사용자 정의 카테고리와 공통 카테고리의 클래스를 **구별하기 어려워**, 탐색 비용 발생
2. 중복 데이터 발생 : 기본으로 제공되는 카테고리 또한 가계부와 N:1 매핑 되어 가계부를 생성 시 공통 카테고리를 가계부와 매핑하여 넣어주어 **공통적으로 관리할 수 없고, 불필요한 데이터 insert 비용이 발생**

해결 시도 1: 클래스 분리

먼저, 사용자 정의 카테고리와 공통 카테고리 클래스 분리를 시도했습니다.

그러나 이 방법은 가계부 내역과 카테고리의 1 : N 매핑 시, 하나의 리스트에 데이터를 담는 것이 어려웠습니다.

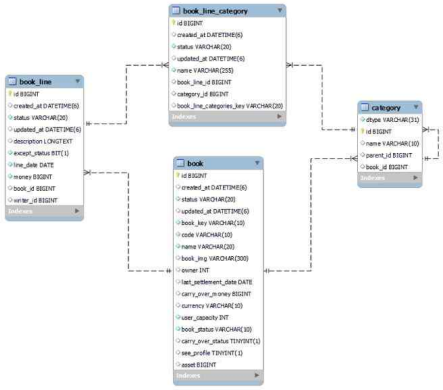
해결 시도 2: 추상 클래스를 상속 받는 Category



상속 관계를 이용해 **공통 자료구조**에 카테고리 데이터를 담아 관리.

→ 공통 카테고리는 단 20개의 레코드만으로 **공통화**를 시킬 수 있었습니다

결론



1. Self-referencing 테이블을 사용하여 계층형 구현
2. 상속 관계를 통한 종류 구별

이를 통해 가계부 내역과 카테고리 간의 복잡한 관계를 명료하게 처리할 수 있었음.