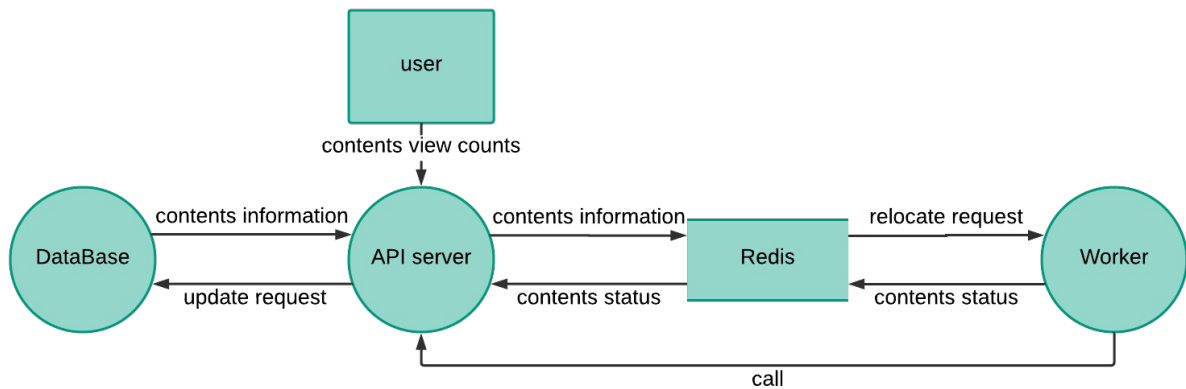


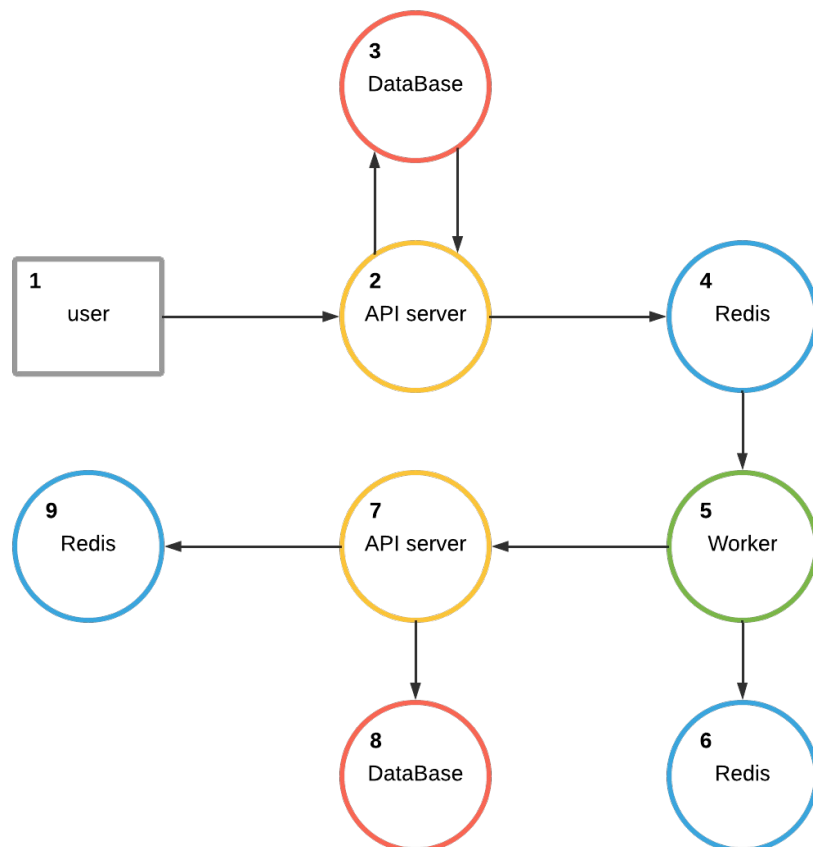
INI Intern Final Project

장예훈 (DB 모듈화) 18.07.27 Tue - 18.08.02 Thu

DFD



Work Flow



User (1)

API server에 cid와 count 정보를 post method로 전달

API server (2, 7)

user로부터 cid와 count 정보를 받아 쿼리 모듈을 이용하여 db에서 필요한 정보를 쿼리 한 후 content의 level relocate 정보에 따라 redis를 업데이트

worker의 호출을 받으면 redis 확인 후 content의 relocate 된 level에 맞게 쿼리 모듈을 이용해 db 업데이트

DataBase (3, 8)

API server에서 인자를 db 쿼리 함수에 넘기면 해당 프로젝트와 연결된 db 쿼리를 실행

Worker (5)

redis의 content status를 확인하여 status가 'update' 인 content의 relocation을 진행

relocate가 정상적으로 완료된 content에 대하여 redis의 content status를 'done' 으로 바꾼 후 API server를 호출

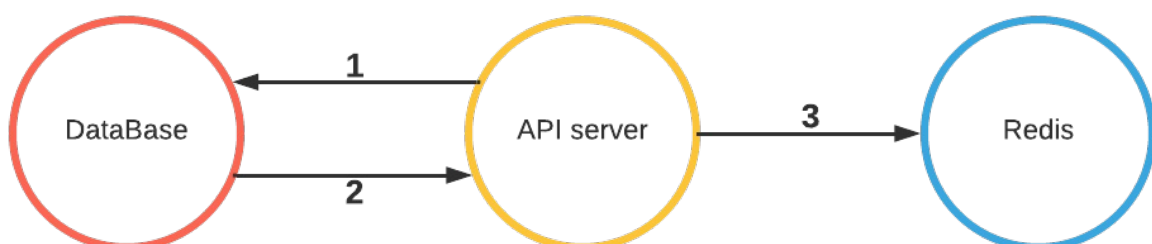
Redis (4, 6, 9)

content count가 업데이트 될 때 마다 올바른 level에 relocate 되도록 content에 대한 정보를 실시간으로 API server와 worker에 제공 (사실상, API server와 worker가 실시간으로 redis를 확인)

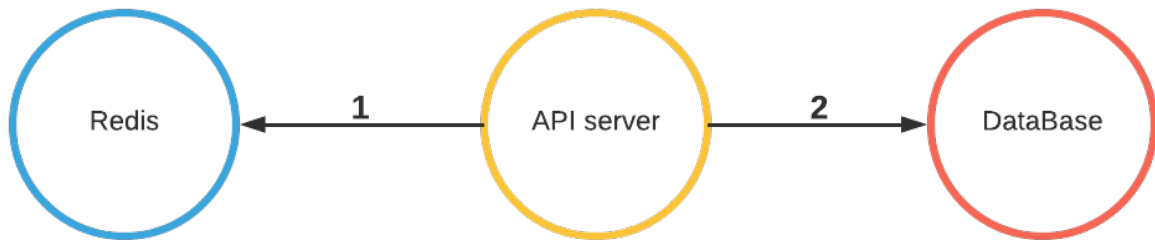
DB Process

Process Detail

redis에 content 정보 업데이트



1. API server가 user로부터 cid와 count 정보를 받으면



Tools

- Ubuntu 16.04.5 LTS
- Python 3.5
- MySQL 5.7.23
- PyMySQL 0.9.2

Module Description

****다시

DB Class - process 내에서 필요한 db sql class 생성

1. select

: 본 프로젝트의 API와 DB 간 데이터 상호 교환 시 **반복 사용되는 MySQL select query 문을 모듈화 한 것**으로 인자는 `table`, `column`, `where_clause`, `order_by`가 있다. `table`과 `column`은 필수 인자이다.

2. insert_contents

: 본 프로젝트에서 콘텐츠가 새로 업로드 될 때 **콘텐츠에 대한 정보를 MySQL contents table에 insert 해주는 기능을 모듈화 한 것**이다. `min_count`가 0인 `level`이 default `level`이 된다. `generate time`은 insert query의 실행 시점이 적용되므로 본 함수는 콘텐츠가 업로드 됨과 동시에 실행되어야 한다.

3. update_level

: 콘텐츠의 view count가 계속 올라가서 특정 레벨로 relocate 되고 api server가 본 함수를 호출하면 **MySQL contents table에서 content_level을 update 하는 query문을 모듈화 한 것**이다. update 하고자 하는 콘텐츠의 `cid`와 relocate 된 `content_level`이 필수 인자이다. 본 함수가 실행되면 contents table에서 `content_level`과 `update_time` column이 update 된다.

DB trigger

- contents table이 update되면 trigger가 작동되고 update_history에 기록됨