

设计思想

数据库采用Redis.

由于要获得用户近期状态更新,所以用有序集合类型进行存储,用所给的unix时间当做score,question_id做为member方便索引用户的动作状态.当需要获取用户状态时需要查找user_id的zset表即可.

当用户获得zset表还需要知道用户问题的action,所以用 user_id:question_id 进行问题动作的查找,找到用户的action,这样就完成了一次查找

对于隐藏问题需要建立另一个键名为 user_id:hide 的Zset,当有问题需要隐藏的时候就把 user_id:timeline 中的对应question_id和time删除,添加到 user_id:hide 中,反之,如果需要显示问题,则进行相反的操作.

数据库设计

Key-Value设计如下:

数据类型	键	值	说明
String	user_id:question_id	action	action为对应问题的动作(1,2,3,4,5,6,7)
Zset	user_id:timeline	time question_id	键中 :timeline 为固定字符串
Zset	user_id:hide	time question_id	键中 :hide 为固定字符串

其中:

- user_id : 用户ID
- question_id : 问题ID
- time : 用户发生动作的unix时间

文档说明

```
.
├── docs
│   ├── README.md
│   └── API.md
└── zhihu
    ├── API.py
    ├── data.py
    └── test_API.py
```

- README.md : 本文档
- API.md : API文档
- API.py : API主程序,完成3个API,分别是user_timeline(),hide_question()和display_question()
- data.py : 生成1000000个键值对存入Redis数据库
- test_API.py : 单元测试

测试环境

- Ubuntu 14.04TLS
- Redis 2.8.4
- Python 2.7.6

依赖

Python需要redis依赖包

调试方法

1. 首先启动redis-server
2. 生成数据写入数据库,如果启动默认数据库设置,则进入zhihu文件夹运行如下命令:

```
$ python data.py
```

如果非默认数据库可设置如下,参数为配置对应的参数

```
$ python data.py -h localhost -p 6379 -d 0
```

-h : 主机地址
-p : 端口号
-d : 数据库

3. 进行单元测试

```
$ python test_API.py
```