Pick Me-数据库表结构

11:10

用户表 用户明星关系表 user_info id user_id 用户id user_id 用户名 user_name 用户 star_id 密码 passport 关注时间 follow_time 应援次数 support_num

动态 资讯 state news state id news id 账号id account_id 明星id star_id 账户名称 account_name 图片 img 内容 content 标题 title 发表时间 create_time 链接 url 图片 imgs 来源 source 来源 source 发表时间 create_time

官方账户offical_account

账户id account_id

账户名称 account_name
明星id star_id

即星id star_id

链接 account url

list_id 榜单名 list_name 明星id star_id 日期 date 排名 rank

评论 行程 帖子 agenda post agenda id 帖子post id 明星id star id 用户id user id 时间 detail time 创建时间 create time 地点 location 标题 title 内容 content 内容 content 类型 type 点赞数 like num

connent
评论id comment_id
用户id user_id
父评论id parent_comment_id
时间 create_time
内容 content
点赞数 like_num
级别 level

明星 star info 用户id star_id 姓名 name 头像 img 微博超话日榜 weibost_daily_rank 亚洲新歌日榜 asians daily rank 微博明星势力日榜 weibostar power daily rank 微信明星权力日榜 weixinstar_right_daily_rank 微信明星势力日榜 weixinstar power daily rank 今日平均排名 average_rank 平均历史最高名次 average_highest_rank 百度指数 baidu_index 当日微博粉丝数 current_weibofans_num 昨天粉丝数 yesterday_weibofans_num 当日ins粉丝数 current insfans num ins粉丝数current_insfans_num

已参加综艺节目数 tvshow_num

在接广告数 ads_num

周边商品 production pro_id 商品名 pro_name

类型 type(代言商品还是同款商品)

分区 数据库 的第1页

后台-接口

2018年8月4日 14:59

1、首页hot推荐

```
要求结果: 所有已关注明星的资讯, 按照时间排序, 50条
请求方式: GET
入参: 用户id
request: {"user_id":userId}
response:
成功:
{
     "state": 10000
     "msg":"success",
     "data":{[
          {"news_id":newId,
          "star_id":starId,
          "title":title,
          "url":url,
          "source":source
          .....
     ]}
}
失败:
     "state":10001,
     "msg":"param is null"
}
{
     "state":"10002",
     "msg":"server error",
}
SQL:
Select *from news where star id in(
      select distinct (star_id) from user_star_relation where user_id = #{userId}
     order by create time desc limit 50;
```

2、某个明星资讯页

```
要求结果:返回目标明星相关的50条资讯,按照时间排序
请求方式:GET
入参:{"star_id":starld}
成功:
{
    "state":"20000",
    "msg":"success",
    "data":{[
        {"news_id":newld,
        "star_id":starld,
        "title":title,
        "url":url,
        "source":source
      },
      ......
]}
```

```
失败:
{
    "state":20001,
    "msg":"param is null"
{
    "state":20002,
    "msg":"server error"
}
SQL:
Select *from news where star_id = #{starId} order by create_time LIMIT 50;
3、饭圈首页
要求结果:返回当前最新的50条帖子,按照时间顺序排列
请求方式: GET
入参: 用户id
{"user_id":userId}
成功:
     "state":"30000",
     "msg":"success",
     "data":{[
         {"post id":postId,
          "create time": "2018-03-20 23:11:23",
          "title":title,
          "content":"XXX粉丝XXXX",
          "like_num":2000
         },
         .....
    ]}
}
失败:
{
    "state":30001,
    "msg":"param is null"
}
{
    "state":30002,
    "msg":"server error"
}
sql:
Select * from post order by create_time LIMIT 50;
```

4、明星个人动态页

要求结果:

}

- 1、显示该明星最近的行程信息
- 2、显示该明星相关的账号所发的状态

```
请求方式: GET
入参: 明星的id号
{"star_id": starId}
成功:
{
    "state":"3000",
    "msg":"success",
```

```
"data":{
         agenda:{
              "agenda id":"002",
              "star id": "20002",
              "detailtime": "2018-08-29 12:30:33",
              "location":"北京南苑机场",
              "content":"XXX",
              "type":0"
         },
         states:[{
              "state_id":9100,
              "account_id":"504"
              "account_name":"吴亦凡官方微博"
              "create time": "2018-03-20 23:11:23",
              "content":"XXX粉丝XXXX",
              "img":img,
              "source":"微博"
              },
         ]}
}
失败:
{
     "state":40001,
    "msg":"param is null"
{
    "state":40002,
    "msg":"server error"
}
SQL:
    1、select *from agenda where star_id = #{starId} order by
       detailt time limit 1;
    2、select * from state where account_id in(
       select account id from offical account where star id=#{starld}
       ) order by create_time limit 50;
   5、明星打榜页
   要求结果:
       1、已关注天数
       2、当前微博粉丝数
       3、全网人气排名
       4、为他应援次数
       5、当前ins粉丝数
       6、历史排名最高
       7、各种榜单排名
       8、在接广告数
       9、已参加的综艺数
   入参: 明星id
       {"star_id":starId}
   成功:
        "state": "5000",
        "msg":"success",
        "data":{
            {
                 "stateId":9100,
                 "starName":"504"
                 "img":"http://...."
```

```
"followDays":1000, //关注天数
             "cunnentWeibofansNum":3444 //当前微博粉丝数
             "currentInsfansNum":5666, //当前ins粉丝数
             "totalRank":12, //平均排名
             "supportNum":2300, //应援次数
             "averageHighestRank":34, 平均历史最高排名
             "weixinstar right daily rank":54,
             "weixinstar_right_daily_rank":54,
             "weixinstar_power_daily_rank":23,
             "weixinstar_power_daily_rank":23,
             "adsNum":20,
             "tvshowNmu":30
             "weiboStDailyRank":3 1, //微博超话日榜
             "weiboStDailyRankDiff":1, //微博超话日榜变化
             "weiboSpDailyRank":12, //微博明星势力榜
             "weiboSpDailyRankDiff":1, //微博明星势力榜
             "weixinSrDailyRank":32, //微信明星权力日榜
             "weixinSrDailyRankDiff":-5, //微信明星权力日榜
             "weixinSpDailyRank":25, //微信明星势力榜
             "weixinSpDailyRank":2, //微信明星势力榜变化
             "asians_daily_rankDiff":43,//亚洲新歌日榜
             "asians daily rankDiff":-2, //亚洲新歌日榜变化
        }]}
}
失败:
{
    "state":50001,
    "msg":"param is null"
}
{
    "state":50002,
    "msg":"server error"
}
Select *from star_info where star_id = #{starId}
Select date from list where list_name =#{listName}
注:
1、平均排名要根据当前的各种榜单排名进行加权计算
2、需要从list表中查找前一天的榜单排名,然后与今日相比得出排名
  变化
```

榜单 list list_id 榜单名 list_name 明星id star_id 日期 date 排名 rank

6、发帖接口

结果要求: 发帖完成后展示帖子页面,包括帖子title,内容content,时间,点赞数,评论数

请求方式: POST

入参:

1、用户id userId

```
2、帖子标题title
    3、帖子内容 content
         "userId":001,
         "title":"我的第一个帖子",
         "contnet": "hello world"
成功:
    "state":60000
    "msg":"success"
    "data":{
         "userId":11,
         "postId":1000,
         "title":"我的第一个帖子",
         "contnet": "hello world",
         "createTime":2018-09-12 12:32, //时间即为创建记录的时间
         "likeNum":0,
         "commentNum":0 //此数据需要从数据库表comment中查找获得
    }
}
失败:
    "state":60001,
    "msg":"param is illegal"
}
{
    "state":60002,
    "msg":"server error"
}
SQL:
//插入这条帖子
Insert into post(user id,create time,title,content,like num)
          values(userId,createTime,title,content,likeNum) //post id自动生成
//查找这条帖子的评论数
select count(*) from comment where post_id = 1000;
```

7、评论和回复评论接口

要求结果:给帖子评论成功之后要显示评论人的用户名,评论内容,时间,点赞数入参:

```
1、用户id
2、评论内容
3、被评论或者回复的帖子id
{
        "userId":10,
        "content":"说得太对了",
        "parentCommentId":1000,
}
成功:
{
        "state":70000,
        "msg":"success",
        "data":{
            "userName":"xiaoming",
            "content":"说得上记去了"
```

评论 connent评论id comment_id

```
msg: success,
     "data":{
          "userName":"xiaoming",
          "content":"说得太对了",
          "createTime":2018-08-04 19:20:22,
          "likeNum":0,
     }
}
失败:
{
     "state":60001,
     "msg":"param is illegal"
}
{
     "state":60002,
     "msg":"server error"
}
```

SQL:

Insert into comment(user_id,parent_comment_id,create_time,cotent,like_num,level) valuse(10,1000,"2018-08-04 19:20:22","说得太对了",0,1);

注:

parentCommentId如果是帖子id,所指向的评论是一级评论,否则是二级评论

connent

评论id comment_id 用户id user_id 父评论id parent_comment_id 时间 create_time 内容 content 点赞数 like_num 级别 level

爬虫数据接口

2018年8月4日 15:58

1、明星信息

参数:

- 1、最近一次更新数据表star_info的时间modified_time
- 2、明星的名字star name

```
返回数据:
```

```
{
    "star_name":"name",
    "img":"imgurl",
    "weibost_daily_rank":"1000",
    "asians_daily_rank":"10",
    "weibostar_daily_rank":"10",
    "weixinstar_right_daily_rank":"122",
    "weixinstar_power_daily_rank":"122",
    "baidu_index":"222",
    "current_weibofans_num":"2222",
    "yesterday_weibofans_num":"122",
    "current_insfans_num":"2222",
    "current_insfans_num":"2222",
}
```

明星信息

微博超话日榜 weibost_daily_rank
亚洲新歌日榜 asians_daily_rank
微博明星势力日榜 weibostar_daily_rank
微信明星权力日榜 weixinstar_right_daily_rank
微信明星势力日榜 weixinstar_power_daily_rank
百度指数 baidu_index
当日微博粉丝数 current_weibofans_num
昨天粉丝数 yesterday_weibofans_num
当日ins粉丝数 current_insfans_num
ins粉丝数current_insfans_num
已参加综艺节目数 tvshow_num
在接广告数 ads_num
更新时间modified_time

2、官方账户

start id如何生成?

参数:

- 1、最近一次更新数据库表offical account的时间modified time
- 2、明星的名字star_name

返回数据格式:

```
{
    "star_name":"name",
    "account_name":"xxx",
    "account_url":"http://....."
}
```

数据处理:将star_name和star_id匹配, 从而匹配账号和明星之间的关系? 自动生成account id? **官方账户 offical_account**账户id account_id
账户名称 account_name
明星id star_id
链接 account_url
更新时间modified time

3、明星动态

参数:

1、最近一次更新数据表state的时间modified time

```
2、star_name
```

```
返回数据格式:
```

```
·
"account_name":"吴亦凡官方微博",
```

"content":"吴亦凡今天....",
"create time":"2018-08-04 13:30:12",

"img":"http://....;http://....",

"source":"微博"

}

数据处理: 通过account_name与offical_account表的账号

名称匹配

动态

state

state id

账号id account_id

账户名称 account_name

内容 content

发表时间 create time

图片 url

来源 source

更新时间modified_time

4、资讯

参数:

}

- 1、最近一次更新数据库表news的时间
- 2、明星的名字star_name (爬虫链接url)

返回数据格式:

```
{
  "title":"易烊千玺...."
"img":"http://"
```

"url":"http://..."

"source":"搜狐娱乐"

"create_time":"2018-07-29 23:20:20"

资讯

news

news id

明星id star_id

图片 img

标题 title

链接 url

来源 source

发表时间 create_time