# 一.开发规范

## 1.后端环境

Python 3.7.3 + django 1.11.8 + mysql 5.5 + vim + redis

## 2.通信协议

http

## 3.通信格式

json

## 4.API规范

一定程度上符合RESTful定义

# 二.数据库结构

**音乐信息表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 作用 | 备注1 |  |
| id | int | 音乐id | 自增主键 |  |
| name | Varchar(100) | 音乐名 | Not null |  |
| Download\_count | Int | 下载量 | Default=0 |  |
| Star\_one | int | 一星数 | Default=0 |  |
| star\_two | int | 二星数 | Default=0 |  |
| star\_three | int | 三星数 | Default=0 |  |
| star\_four | int | 四星数 | Default=0 |  |
| star\_five | int | 五星数 | Default=0 |  |
| star\_avg | float | 平均数 | Default=0 |  |
| create\_time | datetime | 创建时间 | Auto\_now\_add |  |
| update\_time | datetime | 更新时间 | Auto\_now |  |
| url | Varchar(1000) | 音乐资源链接 | Not null | 原始url |

**音乐中间表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 作用 | 备注1 |  |
| id | int | id | 自增主键 |  |
| key\_id | int | 音乐关键字表id |  | 外键 |
| music\_id | int | 音乐信息表id |  | 外键 |

**音乐关键字表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 作用 | 备注1 |  |
| id | int | 关键字id | 自增主键 |  |
| keyword | Varchar（50） | 关键字 | Not null  unique |  |
| create\_time | Datetime | 创建时间 | Auto\_now\_add |  |

# 三.接口说明

1.查找音乐

URL: [http://127.0.0.1:8000/v1/music?](http://127.0.0.1:8000/picture?pic)operation=<?>& keyword=<?>

1.1请求方式1

GET

1.1.1请求格式

json具体参数如下:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| keyword | 关键字 | char | 必填 |
| operation | search | char | 关键字查询 |

请求示例:

0点击搜索之前 交给爬虫

{

‘operation:0,

‘keyword’: ‘关键字’

}

1点击搜索后,先redis没有去mysql数据库获取关联资源名,若该关键字 交给爬虫(ro do 调用爬虫接口)

{

‘operation:1,

‘keyword’: ‘关键字’

}

{

‘type:’music’,

‘keyword’: ‘关键字’

}

1.1.2响应格式

json具体参数如下:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 备注 |
| data | 音乐信息列表 | char | 包含一些音乐信息的列表 |
| code | 状态 | int | 默认正常为200,异常状态再定 |

响应示例:

存在关键字返回的响应

{

‘code’:200,

‘data’:[(id,url,name....),(),()]

}

不存在关键字 返回的相应

{

‘code’:20000,

‘error’:’暂无相关资源’

}